DRAGON USER

The independent Dragon magazine

60p May 1983 Volume 1 Number 1



New Frontiers: Data files, Get and Put

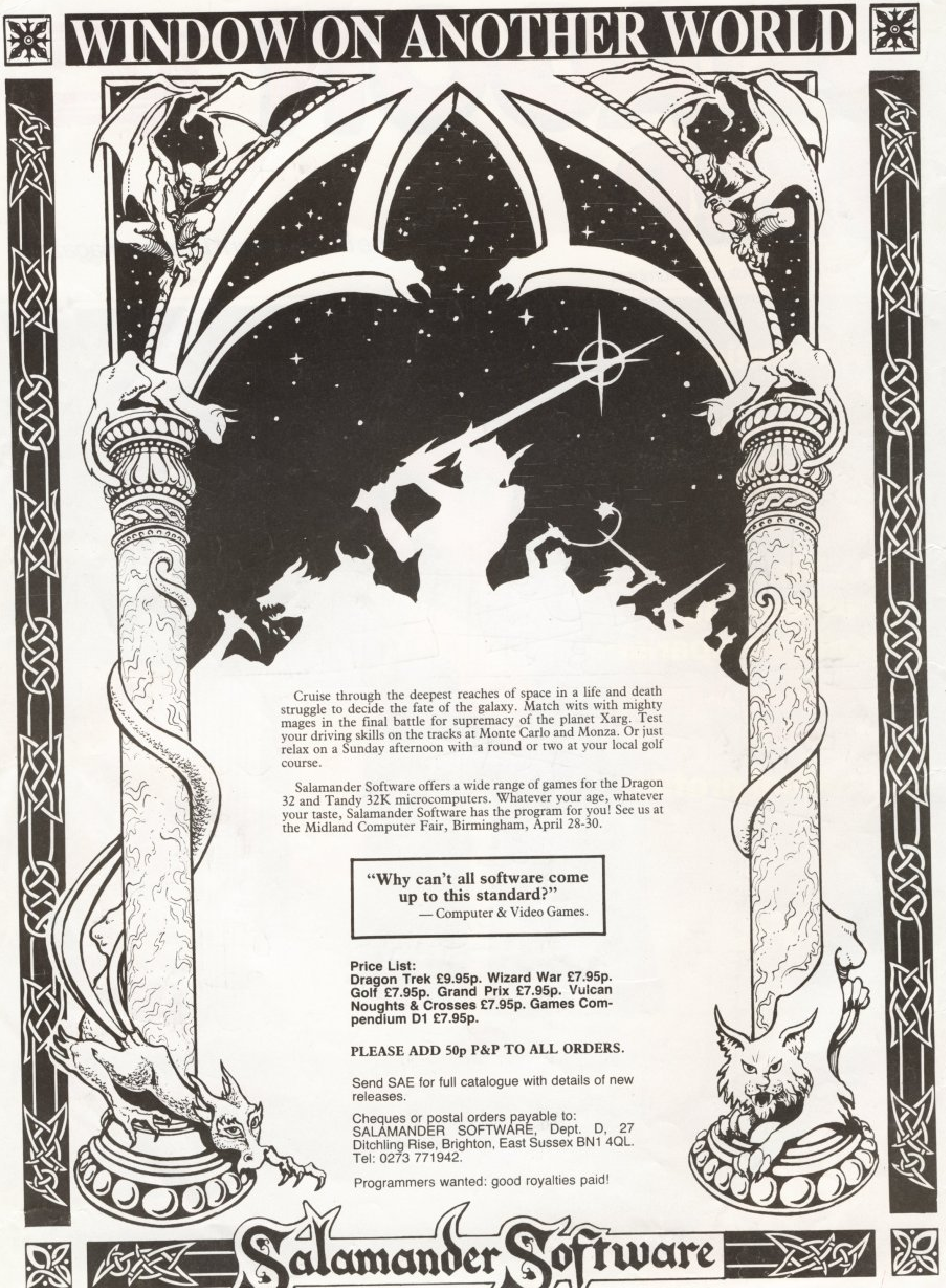
Full software survey

BBC, Vic, Spectrum and Dragon comparison

Moonbase Alpha, Dragonet

Microdeal profile

WIN An Epson MX-80F/TIII printer and word processor.


WINDOW ON ANOTHER WORLD
Cruise through the deepest reaches of space in a life and death struggle to decide the fate of the galaxy. Match wits with mighty mages in the final battle for supremacy of the planet Xarg. Test your driving skills on the tracks at Monte Carlo and Monza. Or just relax on a Sunday afternoon with a round or two at your local golf course.
Salamander Software offers a wide range of games for the Dragon 32 and Tandy 32K microcomputers. Whatever your age, whatever your taste, Salamander Software has the program for you! See us at the Midland Computer Fair, Birmingham, April 28-30.
"Why can't all software come up to this standard?"
— Computer & Video Games.
Price List:
Dragon Trek £9.95p. Wizard War £7.95p. Golf £7.95p. Grand Prix £7.95p. Vulcan Noughts & Crosses £7.95p. Games Compendium D1 £7.95p.
PLEASE ADD 50p P&P TO ALL ORDERS.
Send SAE for full catalogue with details of new releases.
Cheques or postal orders payable to: SALAMANDER SOFTWARE, Dept. D, 27 Ditchling Rise, Brighton, East Sussex BN1 4QL. Tel: 0273 771942.
Programmers wanted: good royalties paid!
Salamander Software

Treble clef babble

DRAGON USER

May 1983

Editor
GRAHAM CUNNINGHAM
(01-839 2449)

Software Editor
GRAHAM TAYLOR (01-839 2504)

News Editor
DAVID KELLY

Editorial Secretary
CAROLINE OWEN (01-930 3266)

Advertisement Manager
DAVID LAKE

Advertisement Executive
SIMON LANGSTON (01-930 3840)

Administration
THERESA LACY (01-930 3266)

Managing Editor
DUNCAN SCOT

Publishing Director
JENNY IRELAND

Dragon User, Hobhouse Court, 19 Whitcomb Street, London WC2 7HF.
Published by Sunshine Books, Scot Press Ltd.
Typesetting by Chesham Press, Chesham, Bucks. Printed by Eden Fisher, Southend-on-Sea.
Distributed by S.M. Distribution, London SW9. 01-274 8611. Telex: 261643

Subscriptions
UK £8 for 12 issues
Overseas £14 for 12 issues

How to submit articles

The quality of the material we can publish in *Dragon User* each month will, to a very great extent, depend on the quality of the discoveries that you can make with your Dragon. The Dragon 32 computer was launched on to the market with a powerful version of Basic, but with very poor documentation.

Every one of us who uses a Dragon will be able to discover new tricks and quirks almost every day. To help other Dragon users keep up with the speed of the development each of us must assume that we made the discovery first — that means writing it down and passing it on to others.

Articles which are submitted to *Dragon User* for publication should not be more than 3000 words long. All submissions should be typed. Please leave wide margins and a double space between each line. Programs should, whenever possible, be computer printed on plain white paper and be accompanied by a tape of the program.

We cannot guarantee to return every submitted article or program, so please keep a copy. If you want to have your program returned you must include a stamped, addressed envelope.

We pay for articles according to the length and the quality — it is worth making that extra bit of effort.

Contents



Editorial

Welcome to *Dragon User* — Britain's first monthly magazine devoted entirely to helping owners of Dragon computers.

It is nearly nine months since we first began to hear rumours that a little known toy company was about to launch a new home computer. At the time Sinclair was having problems with his Spectrum computer but we still thought it unlikely that a newcomer with no experience of the market would be able to break in. Then the Dragon 32 was launched in August. It was undoubtedly a good and powerful machine. We were still sceptical. The documentation was poor and the parent toy company seemed to be in serious financial difficulties.

Dragon Data's Managing Director, Tony Clarke, has, however, managed to pull it off. Various financial institutions bought and re-financed the company which enabled it to build up the production rate to come closer to meeting the unexpectedly high demand; high street retailers and dealers accepted that the machine was going to prove popular and software houses began to include Dragon 32 material in their ranges.

Dragon owners can now look forward to a long relationship with the company. Work is being done to offer upgrades to the Basic and the memory of the Dragon 32. New, more business orientated, computers are being developed at the moment so that Dragon can offer a range of computers of different capacities at different prices, so becoming less reliant on the possible vagaries of one market.

Versions of the Dragon 32 are now being sold in many parts of the world. Later this summer a plant in the United States starts production for the American market. On top of the 100,000 Dragons which will have been sold in Britain by Christmas we can look forward to the additional support of up to 200,000 Dragons in the US.

All this means that Dragon owners can grow with their machines, can bank on the support of any number of software and hardware companies which will, in turn, offer an ever growing range of add-ons and upgrades with which we can continually improve our machines.

In *Dragon User* we will aim to keep you up to date with all the latest developments from Dragon, the latest software, which we will test and rate, and the latest hardware. We will, also, with your help and discoveries, learn how to make the most of the machine we already have.

Useful commands

THE FOLLOWING information about the use of *Pclear* may be of use to other Dragon users.

(1) When the Dragon's manual says '*Pclear* behaves in a similar way to *Clear*' they mean exactly the same way — all numeric and string stores are cleared and arrays 'un*Dim*ed'.

(2) The Dragon's manual also points out that 'As each graphics page requires 1536 bytes of memory only reserve what is needed'. Because the *Pclear* command's lowest number is 1, if your program uses no graphics at all you will waste 1536 bytes of memory ie *Page* 1.

It is possible to overcome the second problem. As Doug Dixon pointed out in P.C.W. Vol 2 No 7 the start and end addresses of Basic are at Peeks 25 to 28. So instead of using *Pclear* use in direct mode *Poke* 25,6 to move the start address to 1537 and *Poke* 27,6 to move the end address two bytes to the right to 1539. Now to check this type

Print Peek (25) * 256 + *Peek* (26)

for the new start address and

Print Peek (27) * 256 + *Peek* (28)

for the new end address. Now you gain an extra 1.5K of memory.

An added advantage is that your program is protected. Try calling any graphics mode and your Dragon will not let you. As this new command is so useful why on earth was the Dragon's *Rom* not designed to allow *Pclear* 0?

Anthony Edwards, Amberley, Berkhamsted, Herts.

Dragon fodder

A 10-YEAR-OLD girl won a Dragon 32 for the school recently. It's a friendly Welsh Dragon, I would like to feed it on best anthracite and good educational programs. The former is easily found but the latter is non existent!

Could you through your magazine ask your readers to send me (or yourselves) details of cassettes, programs etc or at least the names of software houses specialising in Dragon fodder?

You see at the moment I feed the beast by hand and I don't like the glint in its celtic eye!

Save me!

Hywel Thomas (Head keeper), Moorlands Junior, Bath.

P S. I'm sure lots of teachers will read your magazine, so they would benefit from such information.

Word help needed

ENCLOSED WITH this letter is my first year's subscription to *Dragon User*.

One article which would be of great interest to me would be a Phonetic Dictionary for use with "Computer Voice". Personally I have found great difficulty in constructing words. If an article is not possible for some reason then possibly ask readers to submit what words they have constructed and up-date them each month.

J Gale, Wroughton, Swindon, Wilts.

Storing problems...

PERHAPS I can help P A Low (*Popular Computing Weekly* 3-9 March 1983). I too had problems initially with storing programs from my Dragon on my cheap and nasty (£5.00) cassette recorder. I traced the problem to the cassette recorder motor taking too long to reach its running speed, with the result that the first few bytes of the program header were corrupted when trying to *Csave* a program. The difficulty was completely overcome by always starting the cassette recorder manually before pressing *Enter*. There was no corresponding difficulty when *Cload*ing.

The Dragon must have a delay built into its operating system to permit the cassette recorder to reach the correct speed before it begins to send data. In my case, this delay is insufficient, but presumably by *Poke*ing a suitable location it would be possible to increase this delay and ensure error-free storage of data regardless of the cassette motor characteristics, while still using the facility to control the cassette remotely. I would be grateful to hear of any suitable *Poke* values.

For my particular set-up, both *Csave*ing and *Cload*ing are now totally reliable and trouble-free. When typing a long program, I always stop every ten minutes or so to save the work done so far. This ensures that if anything does go wrong, the amount of effort wasted is kept to a minimum.

S J Beal, Addington, Croydon, Surrey.

Solved once . . .

I TOO experienced considerable difficulty in loading and saving programs from a brand new Sanyo recorder, purchased with my Dragon 32.

I discovered after many frustrating hours, that the leads were very badly connected, and after my husband had soldered the connections in the ear, mike and remote jacks for the cassette recorder, it appeared my problems were over. A few months later, the other end (which plugs into the computer) became disconnected and this too, had to be soldered into position. Happily, I now have no problems.

I do feel that this should not be happening and that these leads should be checked before leaving the factory. I now have a little trouble with the connection from the mains to the computer. The plastic was not in the clamp and was causing very bad connections.

M J Marley, Biscovey, Par, Cornwall.

Solved twice . . .

IF YOU are having *Save*ing problems; first put a hand microphone, or other source, into the recorder to test that it does actually record.

By the bye, I would like to obtain the following, but Tandy seem to have taken umbrage at the Dragon's use of the 6809 and are refusing to co-operate. Any suggestions?

(1) Colour Computer Technical Reference Manual (Tandy).
(2) Going Ahead With Extended Color Basic (Tandy).
(3) Basic Microprocessors and the 6800 (By Ron Bishop, Hayden Book Co. Inc).

D L George, Shoreham-by-Sea, West Sussex.

Recorder advice

SINCE BUYING a Dragon in December 1982, I have used four different makes of cassette recorder with it, and found that the best results were obtained from a Binatone Radiocorder (£19.95 from Rumbelows). This is much more consistent than the Sony, Aiwa and Hitachi cassettes which I have also tried — this may be due to circuitry relating to Remote, since some more expensive machines have delay mechanisms to prevent snatching at the tape (which can cause the tape to stretch).

However, even using the Binatone I can't get rid of all I-0 errors and so the following is the procedure I follow when keying a program:

10 Key program
20 N=1
30 Run program
40 If error then debug error: goto 30
50 Poke 65494,0 (In case there was a Poke 65495,0 speed up routine in the program
60 Press record on the cassette
70 *Csave* "Program name"
80 Press rewind on the cassette
90 *Skipf* "Program name" (This is because *Cload* deletes the existing program from store, but *Skipf* doesn't).
100 If I-0 error then rewind cassette: check playback volume: if it was okay, then N=N+1: goto 60 else change volume: goto 90.
110 The save is successful — switch everything off and go for N pints.

The above procedure results in always having a reloadable program, and so no keying effort is wasted.

F McCormack, Langley, Manchester.

P.S. The highest volume I have had N reach is 3 — if it reaches a sufficiently high level to drive you towards alcoholism, it may be better to change your cassette recorder.

No Dragons at Smiths

W H SMITHS has no plans to stock either the Dragon 32 or software to run on it.

It seems both Smiths and Dragon are too busy to consider it.

Smiths' John Roland commented: "What with the Spectrum, Commodore 64 and Oric we have enough on our plate already."

Dragon's Richard Wadman said: "They are interested but at the moment we haven't got the capacity. I think it would be unfair to take on an additional retailer the size of Smiths at this time."

Also, since Boots already stocks the machine, it is not in Dragon's interest to have two high-street stores competing on price.

New factory opens



Production will reach 10,000 Dragons a week

PRODUCTION of Dragon computers is to be stepped up, now that the move to new premises is complete.

The new factory at Kenfig, near Port Talbot, has a much increased manufacturing capacity compared with the old site within Mettoy.

Dragon is now manufacturing just under 5,000 machines a week.

"The move went very smoothly and, over the next couple of months, we shall be stepping up our production to 10,000 units a week," commented Dragon's Tony Clarke.

However, it will take some time for production to exceed demand.

Dragon 32 machines were in very short supply immediately prior to Christmas and all of Dragon's outlets — including the Boots stores — are still receiving a restricted allocation.

"Always when you get a great many orders there is a temptation to throw everything you can out of the door," countered Clarke. "But that is counter-productive.

"All our machines are soak-tested and we have had to adopt a strict allocation system for all of the Dragon's customers."

More micros to follow memory

HAVING sold over 50,000 Dragon 32 machines, Dragon Data plans a formidable hardware expansion in 1983.

First come the discs, planned for the end of April. A single-disc system, with interface and disc operating system, will cost £275.

The package uses a half-height, 5¼-inch single-sided, single-density 48-tracks-per-inch drive with a capacity of 180K formatted (¼M unformatted).

Dragon Data is also licensing the OS9 disc operating system (around £40) and Basic 09, Pascal and Version 7C compilers (all around £40) from Microware in the US.

These will be available in the middle of May, at the same time as the expansion box giving the Dragon 64K Ram. This add-on may include twin RS232 interfaces (not finalised) but should cost around £30.

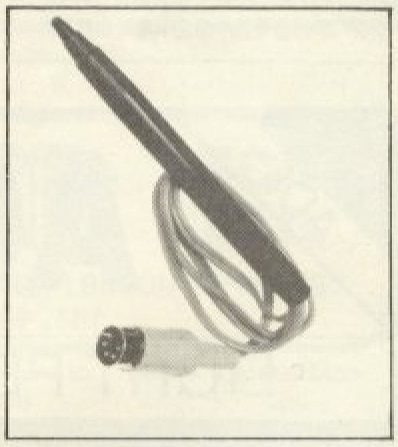
An 80-column card (monochrome only) is in the final stages of development and should be out soon.

Software to run the extensive range of business packages available under CIS Cobol is also planned — to work with programs under 30K in size.

Dragon Data also plans two completely new machines, for launch later in the year.

The first will sell for around £400 and be a competitor for the BBC Model B, the second will be a full-blown business system aimed at the IBM PC/Sirius market.

Trojan lets out light



Just plug into the joystick port

TROJAN Products is now selling a light-pen for the Dragon.

The unit costs £10 and is supplied complete with a cassette giving full instructions and examples showing how to incorporate the input from the pen into a program.

The light-pen plugs into the joystick port on the Dragon and is addressable from the keyboard.

It can be used to input x-y co-ordinate data from the screen and is also fast enough in reaction to be used in certain types of games.

More information from Trojan Products, 166 Derlwyn, Dunvant, Swansea.

Users get options on assemblers

SUDDENLY there is a reasonable choice of assemblers available for the Dragon.

Dragon itself has two versions on the way — one on cassette and one in Rom. Other packages are available from Compusense and J Morrison (Micros).

The two Dragon Data offerings are assembler/editors with a de-bug facility. The cassette-based version, priced at around £10, is the simpler.

The cartridge version will follow soon with more extensive de-bug capabilities and will be priced around £40.

Compusense's assembler is a different sort of package — it is what is called an in-line assembler. Rather than being Rom independent, the cartridge does not interrupt the Basic.

Instead it adds extra commands which allow the assembly language to be added to an existing Basic program. When the assembler cartridge is then executed it runs through the Basic program picking out and compiling the code.

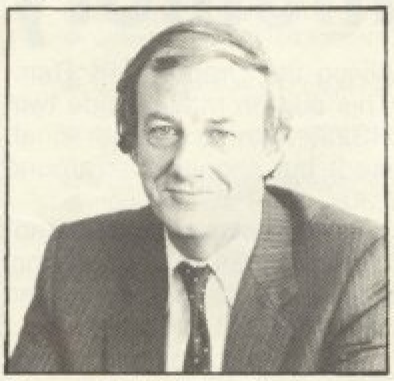
"The beauty of the package," explained a Compusense spokesman, "is that the code is embedded in the Basic. That way there is no need for an editor — the editor is already there in the Basic."

The Compusense DASM assembler is available from PO Box 169, Palmers Green, London N13, price £18.95.

Finally, there is the editor/assembler from J Morrison (Micros). Converted from a 6809 system developed for the SW Tech machine, this is a two-pass assembler which functions independently of the Dragon's Rom.

The package includes error notification (14 codes) and a machine-code monitor. It is supplied in cassette form and costs £27.65. More details from J Morrison (Micros), 2 Glensdale Street, Leeds.

Dragon 32 goes west



Dragon Data's Tony Clarke

THE DRAGON 32 machine will be launched in the US this year.

Three American companies are currently in discussion with Dragon Data with a view to manufacturing the machine in the US.

"Nothing has been finalised yet," said Dragon managing director Tony Clarke. "Whatever happens it will be manufactured over there for economic reasons as a joint venture with a US company.

Decode for Tandy tapes

TAPE software written for the Tandy Colour Computer can be loaded directly into the Dragon 32, using a new cartridge from Compusense.

Both machines run the same version of Microsoft Basic and have similar hardware. This means that a program keyed directly into either machine will run.

However, because of differences in the input/output routines, when a Tandy program is loaded into the Dragon from tape certain of the Basic keywords are misinterpreted.

Within the machine each keyword — such as *Gosub* or *List* — is represented by what is called a 'token'. The problem of software compatibility is that in some cases the same token is used to represent different keywords on the different machines.

The Compusense Decode cartridge simply goes through the Tandy program after it is loaded and makes the necessary corrections to the tokens so that it will run properly.

First the Decode cartridge is plugged into the Dragon. Normally this would interrupt the keyboard input but the Decode pin connector has been specially adapted so that this does not happen.

Then the Tandy program required for conversion is loaded from tape in the normal manner.

Finally, the Decode program is run by typing in the command EXEC &HC000.

When the translation is complete the program, now in Dragon 32 format, can be saved to tape.

The cartridge will successfully convert almost any program written for the Tandy Colour computer in Extended Microsoft Basic. The Decode program has been designed so that it ignores the content of strings.

Machine-code held in subroutines — used to set up the graphics for example — is not affected by the Decode program. Machine-code routines will work equally on both machines without an adaptation.

"The only possible problem could come if someone has used part of the Tandy Rom in a program without using one of the standard machine-code entry points," explained Compusense's Ted Opyrchal. "But in practice professional software houses tend not to access the Tandy Rom.

"We originally developed the Decode program for ourselves and we still use it on a commercial basis."

Decode costs £18.95 and is available from Compusense, PO Box 169, Palmers Green, London N13.

News Desk

01-839 2449



BRIGHTON-based software house Salamander has announced three new titles for the Dragon. These are Starfighter, a machine-code arcade-type game; Super Hangman, featuring high-resolution graphics and a 1,000-word vocabulary; and Sketchpad, a graphics package. More details from Salamander, 17 Norfolk Road, Brighton.

Package additions on the way

IN THE next two months Dragon will launch a substantial catalogue of new software titles.

"We are just finalising agreements for 20 new programs," said Dragon's software development manager, Pam Bolitha.

First comes Rail Runner, a Frogger-type game, followed by El Diablero, an adventure game set in the desert, Whirlybird Run and Shark Treasure.

Computer Voice is a speech synthesiser using string arrays to build-up phonetic codes. These can be used like building-blocks to make up simple sentences which can be incorporated into programs.

Dragon also plans a series of 10 educational programs aimed at the 4–12 age range. The first two titles in this Young Learning Series — dealing with numeracy and literacy — will be available very soon. Four more titles in the series are on the way.

Tandy UK puts Dragon at ease

TANDY has denied claims that it is planning legal action against Dragon Data because of the similarity of the Dragon 32 to the Tandy Colour Computer.

"We have absolutely no objection to the Dragon machine," said Tandy's UK managing director, John Sayers. "We do not have any copyright on our computer in the UK.

"It is true that the two machines are similar in a lot of respects — they use the same or very similar Rom pack, for example — but I can tell you categorically that we are not planning any legal action.

"Besides, by the time these things are settled out of court the machine is often out of date," he added.

Because both machines license the same version of Microsoft Basic and use the same processor some software for the Tandy is directly applicable to the Dragon.

Increasingly, British software houses are looking to the US for Tandy material which can be easily converted for the Dragon. For example, Microdeal is selling material by Tom Mix originally developed for the Tandy. The same is true of Compusense.

Ted Opyrchal of Compusense commented: "Increasingly, the market is moving to produce standard software for the two machines."

Each month we write about one particular club — if you think yours should be featured here write to Clubs Page, *Dragon User*, Hobhouse Court, 19 Whitcomb Street, London WC2 7HF



Doug Bourne (left) and Dave Windle (right)

Two-man band in full flow

***Dave Windle** gives a personal account of the pleasures and problems of setting up a users' group.*

YOU WOULD THINK that being on the committee of two local camera clubs as well as having a full-time job that involves two or three evenings a week plus doing an evening class in computer programming would be enough for anyone. However, when I first approached Doug Bourne with the germ of an idea about starting a Dragon users' group he was immediately keen. As a local magistrate I also have plenty to occupy my time so the decision to form the Dragon Independent Owners' Association was not taken lightly.

Part of our enthusiasm came from Dragon Data's slowness. As you are frantically unpacking your brand new Dragon, impatient to connect it to your television and perform miracles of programming, a small white card flutters to the floor. A few hours later, realising that you really do need to read that manual, you sit back to take stock and come across that card again.

Part of the card is the usual guarantee. Another part invites the new owner to fill in the printed form and become a member of the Dragon Data Club, Dragon's own user group. Unfortunately, at the time of writing, that is the last you will hear of this club.

How many brand new computerists sitting gazing at a faulty listing have wished that there was someone they could turn to? Of course many hardware retailers are very good. They must be very patient, hearing over and over again the same problems which customers find as they unravel the mysteries of whichever machine they are the proud owner of.

So it was in the light of thoughts such as these that the DIOA was conceived. We first considered a local group. A quick survey of two schools (to which we have access) turned up a dozen or so ZX81s, three or four Spectrums, an odd Vic or two and just three Dragons. On the face of it, therefore, a local group was a non-starter. So we decided to take the plunge and began making enquiries to try to guage the interest both among owners and within the mushrooming software industry.

A first tentative advertisement brought around 60 requests for information. Within two weeks this had risen to well over 100 so we felt that the need was there. Many of these requests were accompanied by long letters detailing snags that owners had come up against, mostly with tape recorders, and many complaining about the Dragon manual.

The next step was to contact software houses as we felt that we should offer members more than just a shoulder to cry on. The response to our requests for a discount for members was amazing. Within a few days we received not only many letters offering discounted software but also copies of the software for us to evaluate and eventually review in our newsletters.

A visit to the bank furnished us with much needed financial advice. Though of course banks cannot lend money to start an association such as ours, the manager was most helpful in pointing out some of the pitfalls strewn across our path.

The next problem, as one enquirer pointed out, was one of politics. He was not sure about the ethics of a group run by two people rather than by a committee of members. This is, of course, a very valid point. However, it is a chicken and egg situation — how do you form a committee without having any members in the first place? We decided, therefore, that we would take it upon ourselves to make the decisions and, initially at least, finance the venture.

We felt justified in risking a few pounds on further advertising for members. Our first shock came with the first professionally printed work that we had done. Having designed them ourselves, we were surprised at the bill for our information sheets which was more than double the expected amount. It meant that other plans we had for good quality graphic pads and flashy membership cards would have to wait.

It also meant that our newsletter would have to be duplicated rather than printed. However, we hoped members would be more interested in what we could actually do for them than how showy we were, and in fact this seems to be the case. Our newsletter, *The Dragon's Tale* is, as yet, only one issue old. The first edition was, of necessity, quite skimpy. A short editorial, a few reviews, some details of our discount scheme and a list of software houses filled (almost) the four pages.

We were pleasantly surprised at the reaction of members who wrote and congratulated us on the newsletter. This was followed by a letter from one of the software houses involved in our scheme which also was most encouraging.

To be absolutely honest, we were disappointed that after the rush of enquiries the numbers joining us trickled rather than gushed in but that trickle thankfully continues. On reflection how we would have coped with say 200 members overnight is difficult to imagine. As things have worked out we now have a workable system set up and most enquiries are dealt with within two or three days. New members should get a letter of receipt of subscription within the same period of time.

Eventually we hope that it will be possible to form a committee as referred to earlier. The problems of doing so still loom large — the biggest of these being that members live all over the country and in one case overseas. So, at present, the workload is split between Doug and myself.

Doug is our PR man and also our supplies officer. He was unbearable when he received a large envelope recently addressed to "The Marketing Director". He also has the task of contacting software houses and is responsible for keeping the peace when we step on somebody's toes. Once, full of enthusiasm, we failed to read properly a reply from one of the biggest software houses and involved it in our voucher scheme — which was not quite what it had agreed to. However, to its credit, the firm was very fair with us and in fact, under a different arrangement, has offered our members a larger discount than before.

My job, or jobs, are those of membership secretary and editor of the newsletter. I must admit that I am finding it most enjoyable sifting through the letters, as well as very informative.

The first newsletter was a bit of a one-man band as far as contents go but I am sure that this is going to change. With subsequent issues the aim is to get our members to use *The Dragon's Tale* as a discussion platform and, in my mind at least, it is the most important aspect of our fledgling group.

Already the letters we have received show that out there in micro land are some very active and bright minds.

We hope the Dragon Independent Owners Association will be able to do a lot more for the Dragon owner. We have to accept that we must learn to walk before we run. The days when anyone who had anything to do with computers was an electronics wizard are gone. Many of us ordinary but interested people now make up the majority of owners. We all need help however far advanced we are in our new hobby and that is why the DIAO was started, to try to help other Dragon owners by sharing knowledge and information and by providing a platform for discussion on the many aspects of "taming the Dragon"

Further details of the group can be obtained by sending a self-addressed (stamped) envelope to: DIOA, School House, Nevern Road, Rayleigh, Essex.

Software round-up

*Fighting Samurai and spaceships, manipulating databases and synthesising speech – **John Scriven** finds software that will push the Dragon to its limits once arcade games have lost their thrill.*

WHEN THE DRAGON appeared last summer, it was encouraging to see that Dragon Data was launching software at the same time. All too often, manufacturers produce good hardware but neglect the software side, forgetting that machines that are sold with games potential will not be bought unless the games are available.

Indeed, computers sell to a large degree on the software support available. Since those early days many more programs have become available, and it is interesting to see what progress has been made since then.

Software for the Dragon is available in two forms — Rom cartridges that plug into the side and cassettes that have to be loaded via a cassette player. The material available on cartridge consists mainly of arcade type games. As they are written in machine code, they are fast and some incorporate modifications that make them more fun than the originals.

When you purchase any software, check up to see if you need joysticks. The packaging does not always make this clear, and they are often essential. There are several models on the market of varying quality and price, so if you have not yet acquired a set, it is wise to shop around.

Dragon's own

First of all, the Dragon Data cartridges. Meteoroids is a version of Asteroids, with a few advantages over similar games. It is possible to select the skill level from 0 to 15. It is also possible to use 1,2, or even no joysticks. Controlling the errant spacecraft is fairly difficult until you realise that Newton's Laws on gravity are obeyed rather well, and take this into account. Once you learn to cope with this, it is a lot of fun.

The object is to destroy as many meteoroids as possible before you fall foul of the flying saucers, which are too accurate for comfort. Individual and best scores are shown in a league table. Unfortunately, the display is in black and white in order to use the highest resolution, but this is inherent in the machine and a problem that occurs in many games.

Cosmic Invaders is a rather weak version of the old favourite Space Invaders and is somewhat lacking in sparkle. Anyone who has played the original would soon tire of this cartridge. It is fairly easy to master, and once you discover how to beat it, even the prospect of a high score fails to excite.

Starship Chameleon can be played by one or two players and involves destroying enemy rockets by colliding your own craft with them. The interesting difference is that craft are colour-coded according to whether they are made of matter or anti-matter. If your craft has the wrong matter status (selected with the "fire" button on the joystick) then you will explode. Red missiles that are smart, ie home in on you, add to the challenge of the game. Skill levels from 1 to 9 may be selected and the scores are shown on the screen.

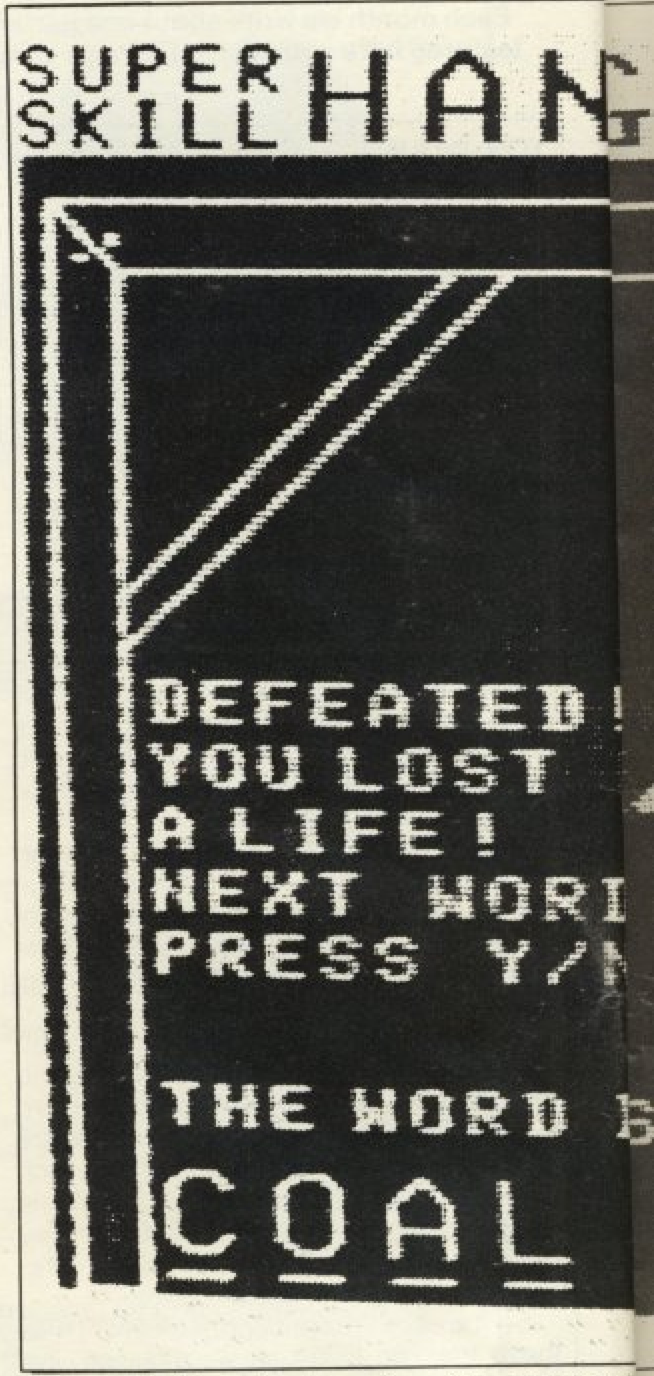
My one criticism of this game is that the background colour is the emetic green that Dragons produce, and with all the high-speed movement on screen you could end up the same colour. On a more serious level, this is more original than Cosmic Invaders, and to my mind, offers much more of a challenge.

There are two cartridges that involve chasing round mazes, although Dragon Data has avoided calling either of them "Taff-man".

Cave Hunters presents you with a cave maze, always the same, with four bars of gold at the base. The intrepid pot-holer has to collect them one at a time and deposit them just outside the entrance. Nothing could be that simple, however — lurking in the cave are unpleasant creatures who devour you when they catch up with you, unless you've recently passed over a power pack, when you have a few seconds to turn the tables on them. This game is fast-moving, entertaining, and more difficult than it at first appears.

Ghost Attack is rather more familiar in design. The object here is to gobble up proton pills littering a maze while avoiding the attentions of three ghosts (unless you've recently passed over an "energiser"). There are three levels of difficulty — "easy", "hard", and "tuff". This game is fun to play but for some undisclosed reason is more expensive than the other cartridges.

Berserk, the last of the Dragon Data cartridges, is based on a popular arcade game that is not often translated to the home computer format. This is another game where colour has had to be sacrificed for high resolution. You control a small man who must be steered through a series of inter-connecting rooms. Robots inhabit the maze and you have to shoot them before they zap you. The walls are electrified, as are the robots, so steady hands are necessary.

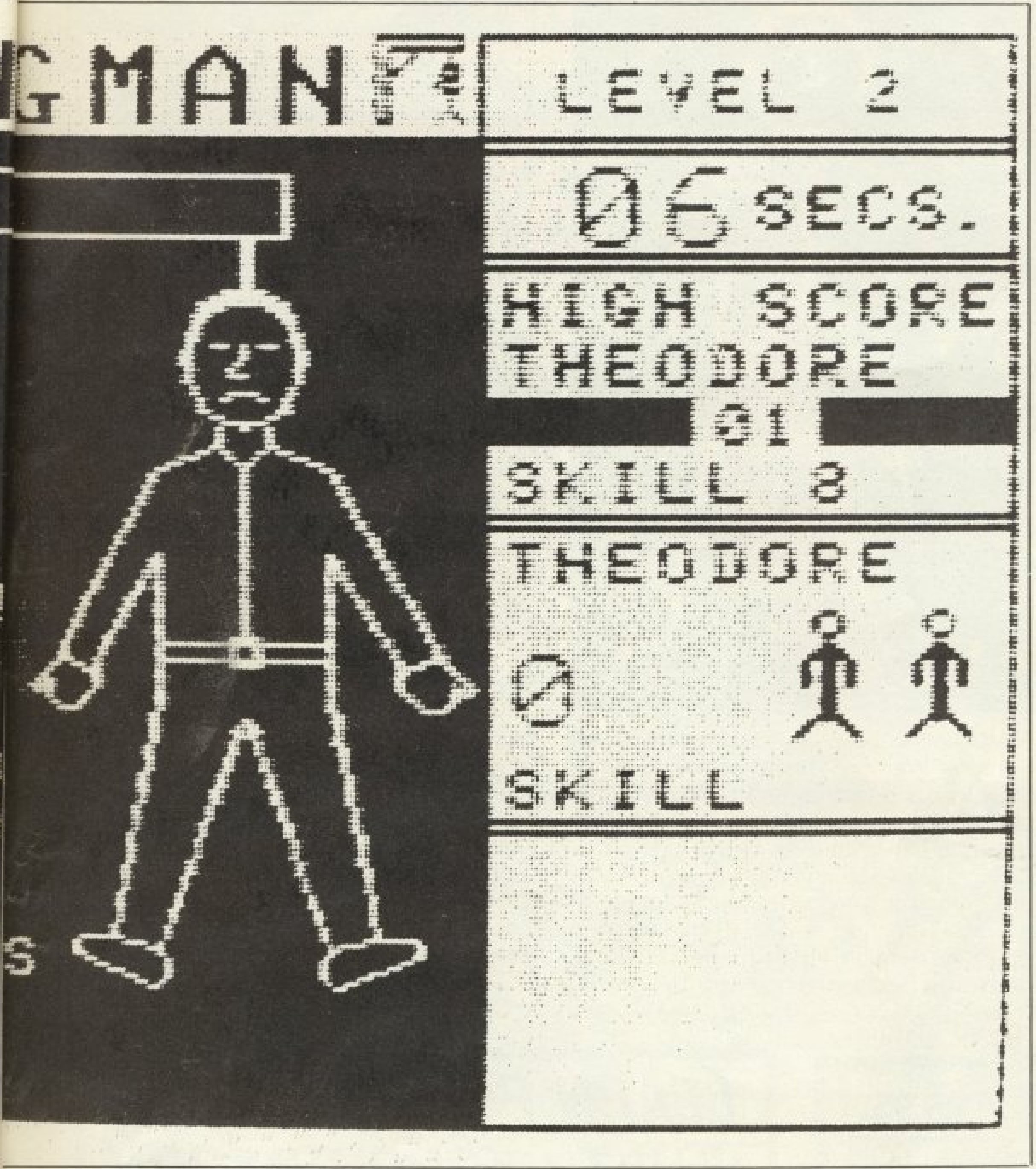


His life in your hands – a timer ticks away the las

As you move off the edge of the screen, another series of rooms appears. It is important to resist the attentions of a lethal bouncing ball called "Evil Orville". He cannot be destroyed and tends to move faster than you can. This is a well-written game and makes good use of the graphics of the Dragon.

The cassettes from Dragon Data cover a wide range of activities and demonstrate the many possible uses for a home computer when arcade games lose their thrill. They are considerably cheaper than cartridges, but naturally take much longer to load and leave you at the mercy of a sometimes unkind cassette operating system. If you've used your Dragon for some time, then you've probably discovered that once the correct level has been set, you have no trouble with your own cassettes, but those recorded on other machines can be temperamental.

Special Selection 1 contains four simple games that tax the parts that arcade games do not reach, namely memory and intellectual skills rather than well-coordinated hand-eye movements. Brain selects two characters from the keyboard



the last seconds as you try to win a reprieve

and gives you clues as you attempt to guess them. Four is a version of Connect Four, which is itself a larger version of noughts and crosses. Horse is a gambling game (in which the computer cheats — I'm sure it does!). Simon is the familiar colour/sound memory game. This tape is good value and even invites you to list the programs in order to discover how the various effects are achieved, so can be used to learn more about the workings of your Dragon.

Graphic Animator uses joysticks to design shapes and pictures on the screen. There is an option to save several pages of graphics and to flick them through quickly to produce an animated effect. This is an interesting cassette, but needs practice to control the movements.

Personal Finance contains three home management programs, Family Budget, Family Accounts and Family Address. The names are self-explanatory, and the programs utilise the excellent file-handling capabilities of the Dragon. Once the main program is in memory, different files can be selected, sorted, changed, deleted, etc, and the new version stored for later use. The documentation, as with all Dragon Data software, is clear and well-written, and the tape is good value if you want to exploit the serious side of your Dragon.

Special Selection 2 also uses the file system to manipulate a database. This program is better than some costing twice as much — even without considering the other programs. It allows you to set up your own database and to put the entries in order, not just in the first field, but in all the others. This means that if you used the system to store names, addresses and telephone numbers, you could sort the file into alphabetical order of names or addresses, or even in the order of the phone numbers.

Of the other programs, Music uses the screen as a sheet of manuscript upon which notes may be written. The tune can be played after completion and stored on tape if you fancy your chances with a new Eurovision song winner. The version of Hangman is extremely feeble — drawing a large Dragon is no substitute for the gallows!

Computavoice is very good value. It allows you to output speech from your television. This has to be entered carefully using phonemes, or sound blocks. If you enter the words as they are spelt, the result is awful, but if you experiment with the instruction book in hand, it is quite simple to make your Dragon offer a friendly word of advice during a game. Longer phrases are possible, but I confess it was two days' work before it said: "Brain the size of a planet and I end up here!" The quality could not be considered anything more than rudimentary, but as an introduction to speech synthesis is has to be good value.

Quest is a cross between Adventure and Kingdom. The aim is simple — to traverse a patch of unpleasant country and invade Moorlock's Citadel. Achieving this aim is considerably more difficult. You need to equip yourself at various markets with men and tools as your progress is slowly revealed on a map of the terrain. My verdict is entertaining — but difficult.

Madness and the Minotaur is in the mould of traditional text adventures. The challenge is to seek treasure in a labyrinth of passages and bring it to the surface. Inhabiting the many rooms are both friendly and unfriendly creatures. Exploring the many locations and learning the different spells is a time-consuming business, but if you enjoy this sort of game, you will not be disappointed with this version.

To the limits

There are now many firms apart from Dragon itself which are offering software. In some cases it is not of a particularly high standard — in others it pushes the Dragon almost to its limits. There appear to be more cassettes on the market than cartridges, so that is where I shall concentrate.

Games Compendium from Salamander Software contains six games. Donkey Derby is similar in concept to Horse on a previous Dragon Data cassette, but has more entertaining graphics. If your steed falls, it lies there with its feet in the air! The game also allowed me to win much more than in real life. The second game, Kingdom, is a management simulation of ruling a country. I did not get on as well as on the BBC demonstration tape version, but this probably says more about my skill in governing than the quality of the program.

Blackjack is the American form of Pontoon, a card game where you try to get a hand worth 21 points. The graphics are reasonable, and the game seems fair, ie I didn't feel the computer cheated all the time! Hunt the Wumpus is such an old computer game, it probably has its origins in valve-driven mainframes. You need to explore a cave system seeking this strange animal, the Wumpus. The computer will tell you if you are getting close and you may fire crooked arrows into the cavern that is his lair. If it's the wrong cave, you may get the arrow on the rebound; if you stumble into his cave, you are eaten — with mustard! Not as addictive as an adventure game, but still a pleasant diversion.

The cassette also contains Noughts and Crosses and Lunar Lander (joystick required). Although no new frontiers are crossed, the cassette is competently written, reasonably documented, and fairly idiot-proof. As with all Salamander software, the cassette comes in a little library case similar to a video cassette holder. ►

◀ For a selection of games, it is good.

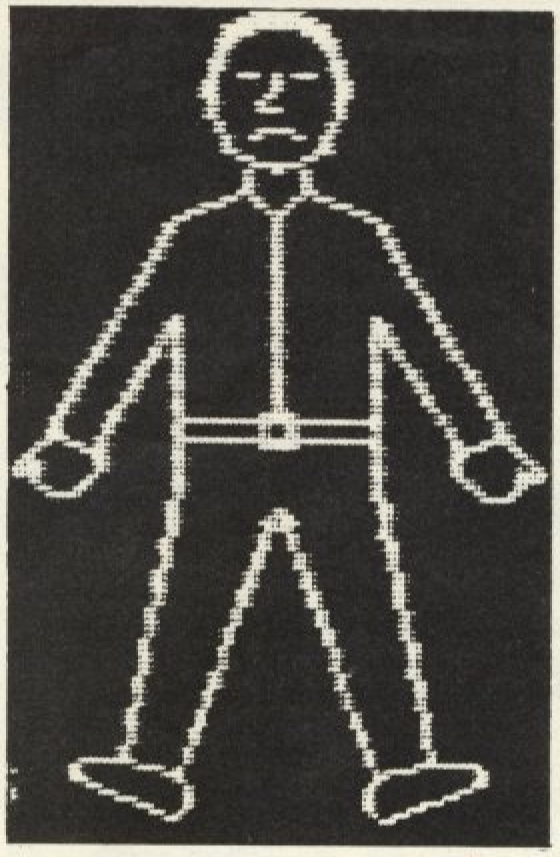
Another offering from Salamander is Golf. The simulation is as close as the Dragon will allow and includes such features as choice of club and strength of shot. The course is shown on a plan, with the various features colour-coded so as to help in choosing options. Just to help the atmosphere (and destroy your nerve) there are additional bits of reality, such as bunkers, trees, wind velocity and direction that tend to interfere with your game.

The graphics are adequate, and this version compares well with others on the market. I think I still prefer the outdoors version, but Salamander's version would do well on rainy days. (I still miss the nineteenth hole.)

Hangman is one of those perennials that is easy to adapt for the computer and that software houses issue when they can't think of anything better to do — at least usually.

Superhangman from Salamander seems to have had some thought and effort put into it. There are three skill levels depending on how much help you want and a vocabulary which won't be quickly exhausted (the computer has a thousand words stored in its memory).

More than that the high resolution man whose life depends on your skill is convincingly done and a timer ticks away the last seconds of his life as you try to win him a reprieve. All in all, if you haven't got it already this looks to be the version to go for.



MC Lothlorien produces a game called Samurai Warrior. This is similar to Kingdom, as it is a game where various choices are available to you, and the outcome depends partly on wise choice and partly on luck. The location is old Japan, and you are among a band of Samurai hoping to achieve a good score. This can be obtained through fighting other Samurai, defending helpless villagers, or by committing Seppuku (the Japanese word for ritual self-sacrifice).

Although the concept is a trifle gory, the game is interesting and well thought out. There are some graphics — Samurai versus bandits, for example, but it isn't as exciting as a Kurasawa movie. I enjoyed the novelty of the game, however, and it certainly made a pleasant change from run-of-the-mill settings of most games.

It is fairly clear to anyone entering a store like W H Smiths that the quality of software packaging is improving all the time. Although this is pleasing to the eye, the contents do not always live up to the expectations conjured up by the superb artwork. It is also true that some excellent programs slip by unnoticed when they are concealed in simple covers.

Microdeal is a case in point. Some of its software is beautifully packaged but proves disappointing, while the least flashy conceals originality. Alcatraz II shows a fugitive from a Minotaur on the cover, but in order to reach this stage you must negotiate prison guards and lasers. This sounds quite inviting until you realise that the game uses low-resolution graphics and, being written in Basic, does not have the speed of a machine-code program.

The concept of the game is interesting but the result is merely average. From the spelling and the fact that the instructions are in a short separate program, it seems possible that Alcatraz II was designed for use with the Tandy Colour Computer.

Planet Invasion, again Microdeal's, ▶

◀ has a gaudy sleeve. This time, the book can be judged from its cover, as this game is a competent version of Defender, complete with smart bombs and high-speed machine-code action. The scene is set rather differently, in that "caloxin crystals" rather than small people are being lifted from the surface of the planet, but the rest of the game should satisfy any Defender addict who wishes to play at home. This game originates in the United States, according to the copyright information.

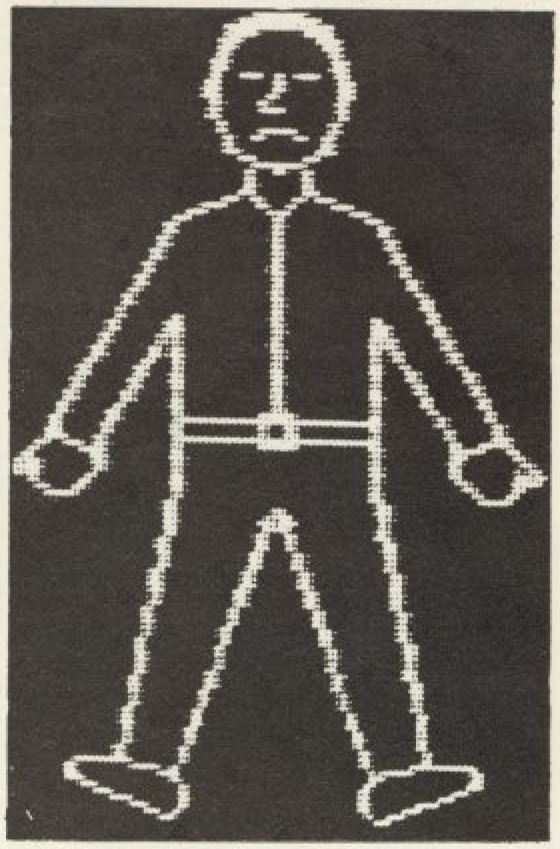
Microdeal's Ultimate Adventure 4 comes in a rather plain cover but inside lurks an entertaining adventure. The locations leap from Arctic wastes to jungle with incredible rapidity, but the program was written by someone with a sense of humour, and is reasonable for a text-only game. It is written in Basic, but this is fast enough for this type of program. My only serious quibble would be with the atrocious spelling — I counted eight mistakes in the instructions alone ("obsticals" being one of the more amusing). This is a high number for a finished product and spoils an original game.

Donkey King is the last from Microdeal and is a "Dragonised" version of an American original written for the Tandy by Tom Nix. The cassette contains not one but three versions of the arcade game whose name differs only in one vowel. True enough the only difference is that of the background/foreground colours but if you choose black and white the resolution is noticeably higher.

When it comes to reviewing straight arcade copies the main criterion must be "how good is this version?". This one lacks the caged gorilla threateningly shaking the bars but does have fireballs and a helplessly screaming Fay Wray.

Despite a practice option I failed to rescue the girl and was regularly crushed to death by barrels and/or fireballs. Written in machine code, the game is fast enough to be addictive and I found that one humiliating failure was quickly followed by another as my finger, acting against my better judgement, insisted on another go.

Forbidden City is an adventure game from Apex Trading of Brighton. The object, as always, is to amass as much treasure as possible without falling foul of the unpleasant things that lurk within the walls



of a city. As adventure games go, I was not particularly impressed. It is purely textual, which means that the descriptions have to be inspiring, amusing or original and here they are barely adequate. It is fun to play, but not really in the addictive class.

Cassette Two is the uninspiring title of a collection of games from Video Productions. The tape seems to be good value in that it contains an index as well as six programs. They are, however, very short, and the graphics are the feeblest possible. With the potential of the Dragon for excellent displays, one wonders why programs should be produced using maths symbols to represent spacecraft. This gives the games the feel of something written for the ZX81. Luckily, the programs improve slightly towards the end of the cassette. Although they are fairly idiot-proof, most users could write comparable games — so why buy this cassette?

Games Pack Two, from Gem Software, contains four games. This collection is far more varied than the last and is far better value. The first game is a flight simulator that enables you to attempt a landing provided you have a pair of joysticks. "Attempt" is the operative word as it is not an easy task. There are no graphics, and the instrumentation is rudimentary, but it is unusual to find so challenging a game amongst a collection.

Hangman, the second program, is a reasonable version with the words stored in data lines, so the program could be altered to contain your own words in a classroom or at home. This is a lot better than the Dragon Data version. The third game is called Speed Boat, the object of which is to race down a river bursting balloons: nothing out of the ordinary, but quite entertaining.

Ships, the last in the pack, is a version of the co-ordinate guessing game, battleships. You have to discover the position of the computer's hidden ships, and if you are successful, there is a display of the unfortunate ship being hit. My criticism of this game is that the co-ordinates have their origin in the top left-hand corner, and the "y" value has to be entered before the "x" co-ordinate, which goes against the accepted conventions. Apart from this, it is an exciting game, and completes a tape that represents good value for money.

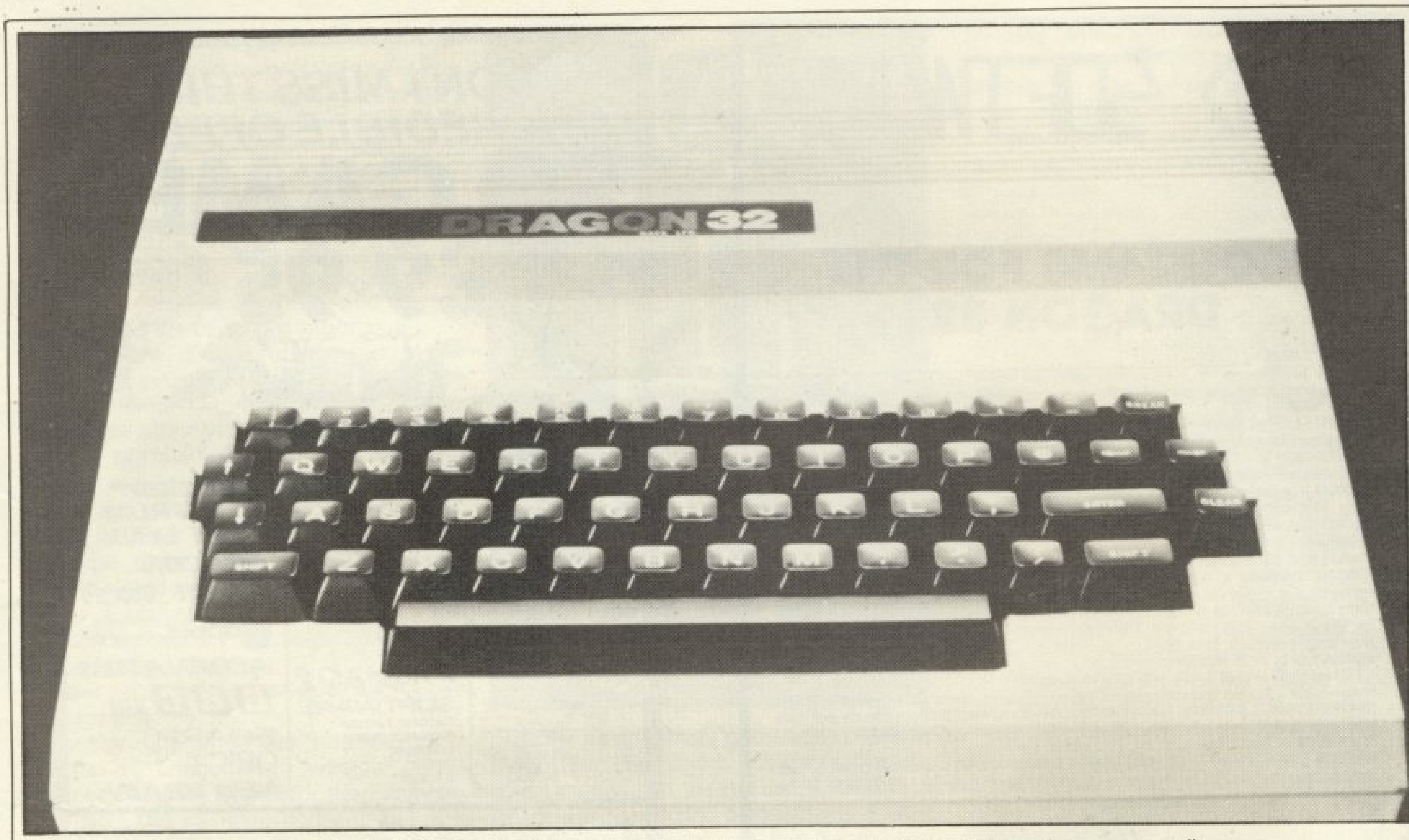
To complete this selection, there is a financial cassette from Hilton Personal Computer Services — Personal Banking System. This cassette contains enough functions to allow you to check on your own bank account. It is menu-driven and can support separate data files. It copes with standing orders and corrections to entries as well as printing statements which you can compare with the bank's.

It could be used by a small business to check individual accounts, but this might be risky — I managed to crash the program twice by entering values outside the permissible range. A program with any business potential would not have allowed me to do this.

To sum up, it seems that there is still room for innovative software on the Dragon. Six months ago, Dragon Data released its first batch of programs, and this is still able to hold its head up in the software marketplace. Of the cassettes and cartridges reviewed here, there seems to be a tendency towards over-charging for products that rarely rise above mediocrity. If the quality of software is not improved, more people will write their own. Ultimately, of course, this would not be a bad thing.

Software packages reviewed this month

Game	*Company*	*Game*	*Company*	*Game*	*Company*
Golf Games Compendium All £7.95	Salamander Software 27 Ditchling Rise Brighton BN1 4QL	Games Pack II £7.95	Gem Software Unit D The Maltings Saubridgeworth Herts	Meteroids Asteroids Cosmic Invaders Starship Chameleon Cave Hunters Berserk All £19.95 Ghost Attack £24.95 Special Selection 1 Graphics Animator Personal Finance Special Selection 2 Computavoice Quest Madness and the Minotaur All £7.95	Dragondata — available from retail (prices shown are recommended only)
Forbidden City £4.95	Apex Trading 115 Crescent Drive South Brighton BN2 6SB	Cassette II £3.95	Video Productions 31 Delmar Road Knutsford Cheshire		
Personal Banking System £9.95	Hilton Computer Services 14 Avalon Road Orpington, Kent	Planet Invasion Alcatraz II Ultimate Adventure Donkey King All £8	Microdeal 41 Truro Road St Austell Cornwall		
Samurai Warrior £6.95	MC Lothlorien 4 Grancy Road Cheadle Hume Cheshire				



The Dragon is one of a new breed of computer designed to last a long time, making it ideal for serious use as well as games

And in the red corner, we have the Dragon

*For some people there is always that nagging doubt. Did I buy the right machine? Or, as is more often the case, was I given the right machine? **Boris Allan** reviews popular home computers.*

IT IS NEVER possible to buy the best computer, because such a machine does not exist, cannot exist. The answer to the question "Is the Dragon the best computer for me?" is also equally uninformative: the answer is "Perhaps, it all depends . . .".

There are quite a number of competitors in the race to produce a cheap, versatile, microcomputer; and, as it is difficult to evaluate any computer in a vacuum, I will try to see how the Dragon 32 compares with three other "home" computers.

The three computers I have chosen to compare to the Dragon are the ZX Spectrum, the Commodore Vic20, and the Acorn BBC Computer; and they have been chosen to exemplify different styles of computing, and different philosophies of design.

A computer can be designed down to a price, that is, the manufacturer starts out with a price, and then builds a machine which can be sold profitably at that price. The Sinclair machines are examples of this "built to a price" philosophy: it is obvious that the prices of £125 and £175 were fixed well in advance of the machine, before it was built, because that was what it was felt that the market could stand.

Another example of the price-dominated mentality of Sinclair machines is the £50 compact disc promised when the Spectrum was launched: the disc has not yet been launched (we do not even know if it is a disc) but it is still priced at £50 — though the disc is now going to have a device to attach it to the Spectrum at some extra cost. The disc price has remained constant, the price of attaching (and therefore using) the disc seems to be increasing.

The Spectrum was also sold for quite a time by mail order only; and this, again, showed the "money at all costs" mentality. There are an untold thousand and one stories (plus) about long waits for the Spectrum, and advertisements being produced offering the Spectrum within 28 days when many people maintain that there were no Spectrums in production.

The BBC computer is, unfortunately, tarred with the same brush: sale by mail order, with no machines to support the advertisements, though in the case of the BBC there were a few in specialist shops. I say "Unfortunately" because the BBC computer was built to a specification, and not mainly to a price.

The specification of the BBC computer is very good, and shows the influence of the accent on the specification being greater than the accent on the cheapness of the computer — though it is an unrealistic manufacturer who ignores the importance of price. It is also an unrealistic manufacturer who ignores the specification to concentrate totally on price.

If a manufacturer is willing to take money on the basis of a promise for the future, we should be wary.

The Vic20 was never sold by mail order; it was sold at first in specialist shops which were Commodore agents, and later in less specialist shops in the High Street. By the time it was introduced into the UK, the Vic had been on sale in other countries, and the Vic was notable for the general lack of bugs when it was first introduced.

The Vic was the first low price colour microcomputer of any importance, and when it was introduced in 1981 it was priced at about £200. This now seems to be quite a high price, and — as the profit margins on computers seem to be rather high — when the Spectrum came along to undercut the Vic, the Vic's price took a drastic fall (as did the price of the ZX81).

How does this compare to the history of the Dragon? The Dragon was notable for being one of the first of the newer, cheaper, computers not to offer itself for months

before its appearance as "available within 28 days" by mail order. (The Vic was probably the first.)

The Dragon was released only in shops, and in quantity: the Dragon appeared in many non-specialist outlets, but to assure a proper back-up it is still wise to try to get a Dragon from a specialist store.

The BBC and Spectrum are similar in that both relied on postal sales, and the Spectrum is notable for relying upon mail order repairs. Depending upon the mail, as they do, might explain why both Sinclair and Acorn seem so slow at communication. The Vic, Dragon, and BBC, can all be repaired in-house by specialist suppliers, whereas the Spectrum is a Sinclair problem.

If the Vic and Dragon are not supplied by specialist retailers, then they too have to be sent away — sometimes the same is true of the BBC.

Different philosophies

I have spent some time on the boring subject of support and sales technique because quite often such things end up being very important: support, and the way the computer is sold, tend to give valuable clues about the nature of the computer's manufacturer.

The philosophy behind each computer is made obvious in many different little ways: the type of keyboard, the container which holds the computer, the type of language, the potential for expansion, and other similar considerations.

Once we start looking at these characteristics we can begin to learn about the Dragon, what it can do, why it can do it, what it is capable of doing.

Start with how a user will approach each of these machines: which of these machines is the most "user-friendly" at the outset, especially for a total novice? The answer has to be, without doubt, the Spectrum.

The most important consideration in the design of the Spectrum (apart from being cheap) would seem to be the initial ease of use. The Spectrum has a form of Basic which is easy to learn, and does not require the ability to spell, because it uses keywords (one key only is used to produce Print).

The Basic has simple graphics commands, which make complicated appearing figures able to be drawn with ease (as long as one only uses one Ink colour).

A computer which is very easy to use at the outset, is not always a computer which is easy to use when one wants more demanding things. All the three other computers have proper keyboards, though at the outset it might be more difficult to remember how to spell Print (or that on the Dragon colour is spelt Color) ultimately it will be found that keywords are very restrictive, and that the dinky touch-sensitive keyboard on the Spectrum is a pain.

If a computer is designed to be easily extended, then it must have a moving-key board, and not have to depend on private suppliers to produce proper keyboards. An industry has grown out of the deficiencies of the ZX81 and Spectrum. Of the three proper keyboard machines, I have found that the nicest keyboard is that on the Vic, though none of the three is poor.



The BBC micro – problems upgrading from Model A to Model B



The Sinclair Spectrum has the least built-in potential for extension

When we move to the languages available on the machines, at one end is the Spectrum and Vic, and at the other end is the BBC and the Dragon. Though the Spectrum (like the BBC and Dragon) has special graphics commands, the language as an entirety is limited, and the Vic (though a colour computer) does not have in-built high resolution commands — though by buying a special Rom pack it is possible to add such commands to the language.

With the Spectrum it is almost impossible to add new commands, partly because of the extreme dependence in Sinclair Basic upon the use of keywords.

Both the BBC and the Dragon use extended Basics and the languages have potential for expansion. The Dragon's extended Basic is not as powerful as the BBC Basic developed by Acorn, but given the content of most of the books published so far about the BBC computer the power of the BBC's Basic has not been recognised or used.

Where some of the books on programming for the Spectrum and Vic are pushing the use of their Basics to the limit, in the case of BBC and Dragon books this potential is still dormant. As the BBC and the Dragon are both more complex machines, the manufacturers themselves are probably not aware of the full potential of their own machines.

As the Spectrum and Vic are inherently simpler machines it is far simpler to cover a larger area of knowledge about the machine: this is partly why the manual for the Spectrum is so much more comprehensive than most other manuals. One has to be very keen to get beyond the information in that manual.

The same is true, but less so, for the Vic: the Vic developed out of the Pet/CBM series of machines, and that Basic has been very successful over a series of years. When a colour graphics language is developed it makes sense to try to develop special graphics commands as part of the language, and the Basic for the Vic did not make sufficient movements towards graphics, eg the lack of high resolution graphics commands.

BBC Basic is, as I have noted, a potentially very powerful language — used to its potential — and though Dragon ▶

◀ Basic is less powerful in some respects, some of the facilities on the Dragon are superior to those on the BBC.

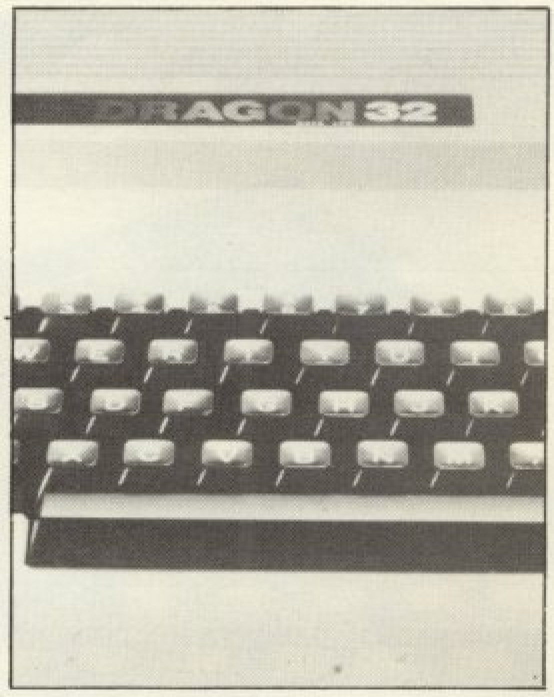
Commands such as Play, or Circle (which can also draw ellipses) are very helpful, though setting up a loop to play the same tune at all 255 different speeds is very wearing on the ears.

A rather surprising omission in the Dragon's Basic is the lack of a facility for forming user-defined characters (a facility available on all the other three colour computers). David Lawrence (Chapter 5 of *The Working Dragon*) gives routines to overcome the deficiency.

In terms of the loading and storing of the Basic programs you have written, the Vic and Spectrum come out best — possibly because both are designed for the utter novice, or possibly a variation in emphasis. The use of cassette recorder on both Vic and Spectrum is easy, though easier on the Vic than it is on the Spectrum (because with the Vic one has to buy a special Commodore cassette recorder).

On the Dragon and BBC the connection between computer and cassette recorder is by Din plug connected to three jacks: it is possible to control the cassette recorder from the computer (as is possible with the Vic). The Dragon's commands Motor On and Motor Off are useful extra facilities: on the BBC you have to pull out the Remote jack.

For somebody such as myself, I find the availability of the Tron (trace on) and Troff (trace off) to be highly useful. Tron outputs on the screen the number of each statement as it is encountered within the program. Again, as a person used to more sophisticated systems, the renumber command in Dragon Basic (Renum) is very helpful: these are the type of commands normally provided by toolkits which are sold as extras on other computers.



The Dragon – its chip wins

In the case of the Vic and Dragon it is possible to plug in cartridges which run instantly when the computer is switched on. The use of cartridges means that the Basic system is frequently switched off, and other languages can be used with the least difficulty; to load a language and compiler into Ram uses up space, and thus restricts the room available for programs.

On the BBC computer it is possible to insert Rom chips, with la. guages or utilities such as word processors, but this is rather more tedious, and less flexible (and more likely to lead to bent pins on chips). The Spectrum has no facility (in the design) to use Rom-based languages and utilities.

The potential for improvement for the Dragon is greater than the potential of the basic machine in the case of the other three computers. The reason why this is so, is that the microprocessor for the Dragon is the Motorola 6809, whereas for the Spectrum it is the Zilog Z80, and for the other two it is the MOS Technology 6502.

Over a wide range of benchmarks, running machine code, the 6809 was upwards of 1½ times faster than the Z80 or 6502; and 6809 Uniflex Pascal (Pascal, like Basic, is a programming language) was only 1½ times slower than a Z80 using assembly language. (68' *Micro Journal* Vol 3 (4)).

The Dragon is, then, a potentially powerful machine: if it is possible to use a common language such as Pascal, and be only slightly slower in running than a machine code program on other computers, this gives the user great power. It is far simpler to write a program in an ordinary language such as Basic (or Pas-

cal) and takes much less programming time, so that if the resulting program is not that much slower there is no need for writing in machine code.

When we talk about the basic machine, we realise that few computers ever stay as the Basic machine, and become extended in many directions. An important point to consider is thus the ways in which a computer can be extended.

The computer with the least built-in potential for extension is the Spectrum, and that with the greatest potential is the BBC Computer (Model B). At the moment the computer with the greatest range of peripherals is probably the Vic, partly because it has been around for the longest, and partly because it was designed to use many of the peripheral devices available for the Pet/CBM series.

Though many private firms have produced extras for the Vic, Vic owners are encouraged by Commodore to buy Commodore peripherals — starting with the obligatory special cassette recorder. In terms of its initial design, though, the Vic is limited.

Considering that the Spectrum has now been with us for quite a time, there has been little produced to expand its potential. By "expand its potential" I do not mean the production of proper keyboards, or sound boxes to amplify the Spectrum's tiny chirrup, but items such as discs. Most of the expansion of the Spectrum can be seen to be to remedy deficiencies in the original Spectrum: buying a soundbox and keyboard (to try to attain the facilities of the Vic or Dragon) will bring the price above £200; mind you, to increase the size of memory on the Vic to any decent amount is also going to be expensive.

Moving up

One thing there is for the Spectrum, are games galore: however, for how long will the Spectrum be Sinclair's main machine? Will there be any easy way of upward compatibility? (Tandy's TRS-80 series are very good at assuring upward compatibility.)

The BBC has the greatest potential because it was so designed, but even then there have been problems with the upgrading from BBC Model A to Model B. With the up-coming "Tube", the BBC computer will be able to use other microprocessors (such as the Z80 or 6809, or the NS16032) as well as other devices. The BBC was produced before the Spectrum appeared, but it is only now that discs have become readily available. The BBC is a proper machine designed to be of use for quite a few years; the Spectrum, one feels, is not intended to have such a long life: though some think that the ZX81 is a disposable computer, the Spectrum is still too expensive to have reached that stage. How long will the keypad last?

The Dragon is one of a new breed of computer, computers which are not designed to be this year's computer, but designed to be of longer validity. On those grounds alone, of the four machines I could only recommend the BBC or the Dragon for anything which pretends to be a serious use. So how have I found the Dragon performs when I am using it?


Increasing memory size on the Vic is expensive

I hate computers without an On/Off button, I find something primitive about having to unplug everything, or to pull out the jack, so I was pleased to discover the button at the rear. I was also pleased to find that (unlike Acorn computers) I did not have to buy a lead to connect my computer to the cassette recorder. I like the reset button, as well as the break key, and was glad that when I reset the system I did not lose my program (to be recalled by Old, as has to be done on the BBC) Switch on and wait.

I get the heading, and start typing: nice feel to the keyboard. No lower case, pity, it can tart up printed output quite nicely, but who really cares? The colour wavers after a while and goes, and I wonder if this is anything to do with the fact that the sound comes out of the television speaker. The reason why I wonder this, is that the same happens on my television when I use a Vic: the Vic also sends sound to the television speaker.

The problem is definitely due to my television because I have used Vics which are well-nigh perfect on other televisions, but they collapse into black and white on mine. The colour of the Spectrum and the BBC is always good on my television, and both those have internal speakers (and little else in common).

Having made my first mistake, I have an error message: an incomprehensible ?SN at line something or other. Looking at the list of errors I find I have made a syntax error, and given time I expect I would be able to decipher most of the error codes at sight; and as there are 24 errors which can be reported I reckon that error-reporting is quite good. But what was the error?

I am used to typing in programs where spaces are not significant (ie spaces between letters do not matter). Searching through the Dragon manual I could not find where it was explained, but it seems that in Dragon Basic, spaces are sometimes needed, and sometimes they are not — my error seemed to be in a For loop. It took quite a time before I realised that this was what was producing the error, and the manual did not help greatly here.

Reading the section on For Next loops (p. 56) I came across the interesting line

```
20 FORI=AtoBSTEPC
```

where it was not too clear where there were spaces, and the "TO" part of the statement was printed as "to".

What the Basic interpreter was doing was taking a line FORI=NTOM as equivalent to FOR I = NT with the rest of the line missing (Dragon Basic thinks that the variable NTOM is NT, it forgets all letters after the first two). This should be made much clearer.

The Dragon is in good company (eg the Atom and the IBM Personal Computer) in being fussy about spaces, but it should be made clear. What makes it more complicated is that sometimes spaces do not matter.

As a machine the Dragon has great potential but to use that potential the manual should have some explanation of how memory is set out, and to find that out needs reference to the four sides called "Additional information". Even then the information is fairly uninformative. A further useful item is a list of useful locations, such as those in the helpful sections of the Spectrum manual, or those in the Apple II documentation. Others, less bigoted than myself, found the manual quite helpful (but in time . . .?).

Having to edit out my syntax error, once I had found what it was, came very easily with the Dragon's Edit command, and after some experience with the editor, I was very impressed. I was also very impressed by the need for a repeat key.

With the 6809 chip being the best 8 bit chip available, I am looking forward to trying out languages such as Forth, which are perfectly suited to the chip's power and capabilities. All in all, the Dragon is a machine with potential as well as present attractiveness.

Acknowledgements

I would like to thank NSC Computers, 29 Hanging Ditch, Manchester (061-832 2269) for their help, assistance, and advice.

Active Software.... "SECOND TO NONE"

" The best buy must now be the recent Games Tape 1 from a new source - Active Software - which has taken some established ideas and implemented them beautifully using all of the Dragon's facilities cleverly. There are eight games, several of them as good as those offered elsewhere as individual cassettes." (YOUR COMPUTER — APRIL 1983)*

"Better than any I have had before." W. PATTON (Cheshire)

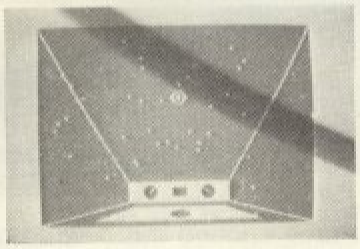
"This tape is by far and away the best we have. You are to be congratulated on the end product." M. RAVENSCROFT (W.Mids)

"I have received immense enjoyment from each of your games without exception." B. HUMPHRIES (S.Wirral)

8 Superb full length programs on one cassette. An incredible 200K of impressive programming, spectacular colour graphics and sound effects!

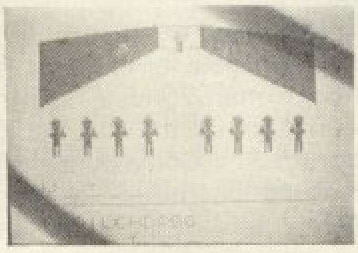
Interplanetary Trader

A massive 30K of real time adventure trading in the year 2283. Encounter the hazards of space flight: pirates, meteor storms, energy loss, etc. Comprehensive on-board computer gives status reports, damage reports, warnings, etc. Definitely a challenge.



Execution

Superb hi-res moving graphics game. Play the computer or an opponent. Features inexhaustable data base, continuous scoring, overall rating, 1-4 players and good sound effects. Hours of fun guaranteed! A massive 30K.



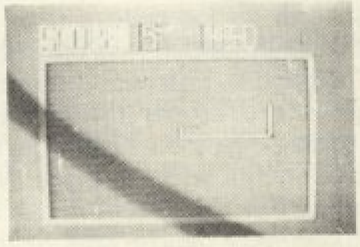
Wumpus Mansion

Incredible fun! Outwit the Wumpi (if you can . . .) and raid the mansion. Collect treasures, and Wumpus Blasters and escape through the maze of tunnels, but don't trigger the TIME BOMB. Different each time you play. Uses 29K!



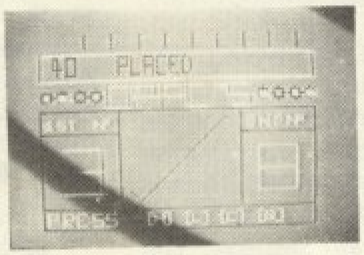
Wipeout

Challenging game of fast reaction that begins easy but gets progressively harder. 1-4 players, on-screen scoring, good sound effects and 4-colour graphics. Be warned — this is not easy!



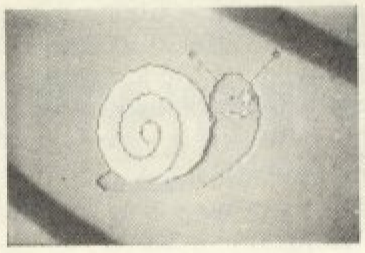
Hi-lo

Is it skill or luck? Turn £50 into £1,000,000 with this simple, yet highly-entertaining game. Brilliant sound effects and 4-colour hi-res graphics. Uses over 20K!



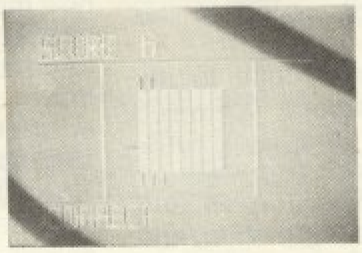
Snail Pace

The graphics are excellent — the excitement is almost unbearable. 1-8 'racing snails' whizz up the racecourse to the finishing line. Computer calculated odds and form that really works. A superb piece of complex programming ensures hours of entertainment. A massive 26K!



Atom Hunt

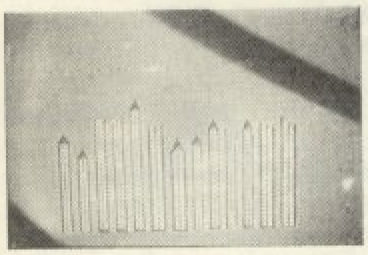
95 per cent full-colour graphics! Another superb feat of programming to stretch your mind and your Dragon's memory chips to the absolute limits! 1-4 players, individual ratings. Includes a unique repeat game facility. Uses over 20K!



Air Assault

A simple yet addictive game in full 4-colour hi-res graphics. Attempt to blast the skyscrapers to make a space and land your crippled helicopter. Uses sound and realistic movement of helicopter.

COMPREHENSIVE 3-PAGE INSTRUCTION SHEET IS INCLUDED



GAMES TAPE 1

8 game cassette. Just:

£6.25

which includes postage and an UNCONDITIONAL REPLACEMENT guarantee.

JOYSTICKS NOT NEEDED

In the unlikely event of our product failing to load just return it to Active Software for an immediate replacement.

Why not put a first class stamp on your order and we will despatch your cassette within 48 hours of receipt.

Active Software
117, Ickneild Street,
Birmingham
B18 6RZ.

*compendium category

ALL ORDERS DESPATCHED WITHIN 48 HOURS OF RECEIPT.

Post coupon now to:

Active Software
117 Ickneild Street,
Birmingham, B18 6RZ.

Please rush me a copy of GAMES TAPE 1

I enclose a Cheque/P.O. for £ ________

Name: ________

Address: ________



John Symes: "A good Dragon program could be converted to Tandy and marketed in America – the royalties could be considerable"

Microdeal fills software gap

*Software Editor, **Graham Taylor,** talks to **John Syme** of Microdeal, one of the first firms to develop a full range for the Dragon 32.*

FOR TWO COMPUTERS that appeared in significant quantities at approximately the same time it is surprising how different is the state of the software markets for the Sinclair Spectrum and the Dragon 32.

The long established ZX suppliers took only a few months to get to grips with the new machine and began to produce interesting new programs that made full use of its potential.

Despite being well received by the computer press and being the one big sales success of the parent company Mettoy the software market for the Dragon has been both small and uninspired.

Many of those who bought their Dragons at Christmas and who are not yet skilled enough to write their own programs were disappointed.

One of the companies trying to transform this situation is Microdeal, a St Austell-based software house which specialises in Dragon software.

Microdeal began as a one-man operation with company director John Symes distributing Tandy Colour computer software from home; "We decided to take on a new computer and the Dragon sounded interesting," explained John.

It is tempting to suppose that just as Spectrum software could develop naturally from the ZX81 so the Dragon could depend on the constantly emphasised link between itself and the Tandy as a similar basis for development. According to John this link is more fiction than fact; certainly a good deal of conversion is required. "Ninety-nine per cent of the material for the Tandy colour won't work on the Dragon."

This is a fact important to John since at the moment nearly all the games he markets are converts from Tandy originals.

Apart from the technical difficulties there are other problems. It certainly isn't a question of converting everything available for the Tandy in America into programs for the Dragon over here; there are important differences in the markets.

"In a way we're selling an American product to English people; it's important to remember that American programs are written for American tastes — they don't like complicated adventures for example and are very much more add-on minded.

"There are a lot of things like Rom packs for programming Epsons that we could adapt from the Tandy but the market isn't interested."

It was again a question of adapting a Tandy original with their first significant serious program the "Telewriter" word-processor; John is very proud of it. "Telewriter is our first-ever business program; it proves the Dragon isn't just a toy."

Although the "Telewriter" looks like being a success it seemed to only emphasise the question I implicitly raised at the start: "Why do we have to rely on a converted American original — where are all the British Dragon programs?"

John shrugged his shoulders: "What little we've seen in Britain just hasn't been good enough. I think it's partly because the 6809 is an unknown processor over here; those who understand machine code have often learnt it in college — that usually means a Z80."

John also lays part of the blame squarely on Dragon itself. "Dragon has been of no help whatsoever to any of the software houses; they didn't even tell us they had reconfigured the Ram — it meant we had to withdraw two games.

"People bought games after playing them in the shop but found they didn't work on their machines at home — naturally they assumed the machine was at fault — it must have cost Dragon a lot of money in unnecessary repairs."

Whatever the initial setbacks Microdeal is now doing well, although John was reluctant to go into details: "Let's just say we are meeting the market demand."

Microdeal currently has four staff and may take on more to cope with the demand. "Up to a month ago it was virtually running from home but we needed to get a 24-hour turnaround."

John was keen to stress the importance he places on speed. "Orders on my desk by nine were packed and ready to go by ten-thirty. To be honest mail-order is a pain with all the time it takes up but it's necessary until the dealer network builds up."

Now that Dragon software is starting to be produced in substantial quantities building up a reliable dealer network is obviously a major objective but it has to be done carefully. "We won't let just anyone sell our stuff — people have to be able to get the correct back-up advice and information — most prospective distributors are turned down."

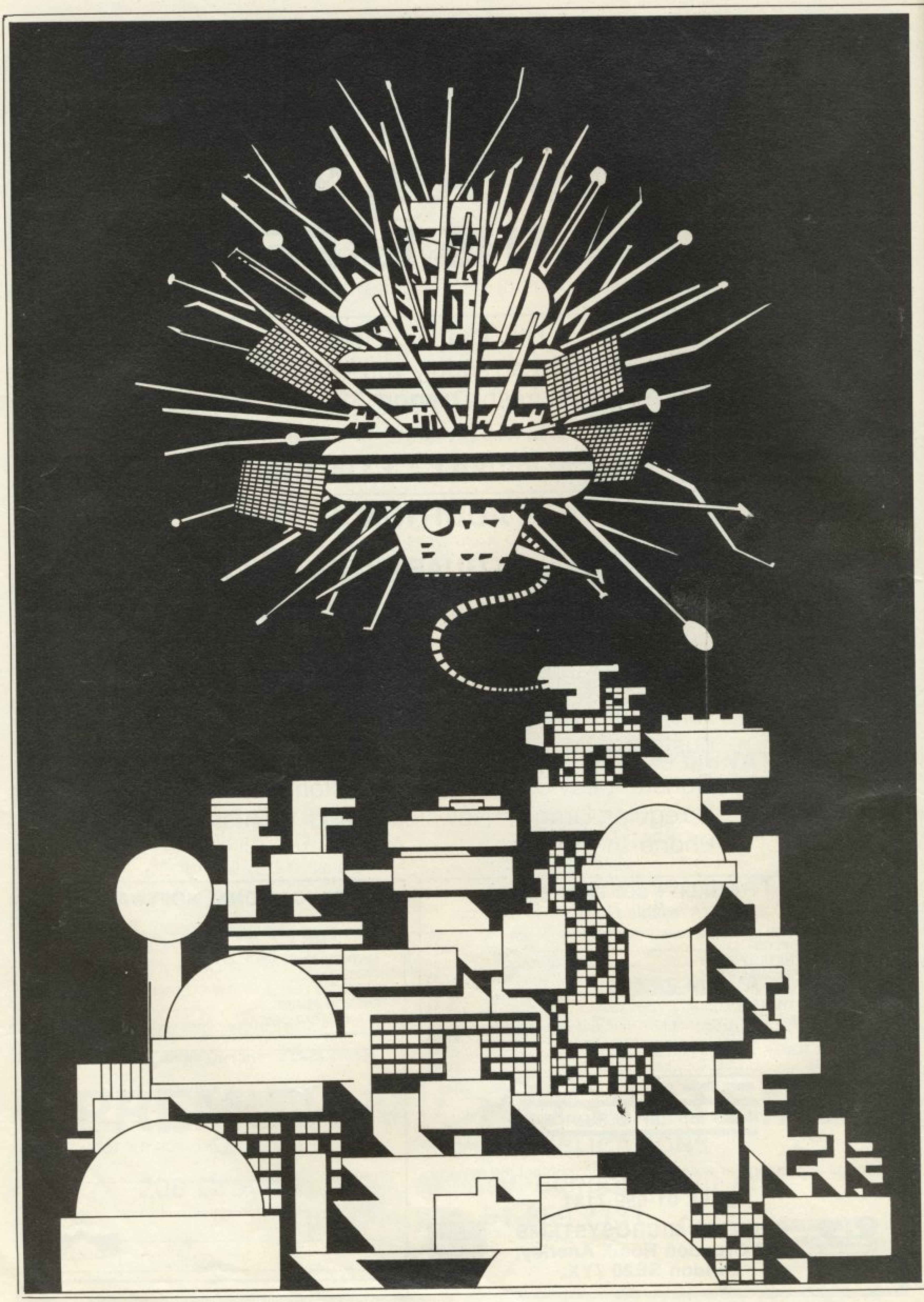
His concern that as the fledgling market is developing the retailers should be protected is reflected in his uncompromising attitude towards software libraries. "We don't let them have anything — how can the dealers survive if we do?"

Perhaps as a sign that British programmers are finally waking up to the potential the home-grown machine represents, Microdeal is soon to produce its first game from a local programmer — a version of backgammon utilising a light pen.

Certainly John hopes it's only the beginning. "It isn't from choice that I'm paying thousands of dollars in American royalties I'd much prefer to see British programmers getting their act together.

"If anyone has got a decent Dragon program they'd be crazy not to do something with it.

"Remember the whole process could work the other way, a good Dragon program could be converted to Tandy and marketed in America — the royalties could be considerable."



Landing on the lunar hover pad

***David Windle** introduces a new game for the Dragon 32 which uses the high level graphics commands to draw the lunar landscape.*

IMAGINE YOU ARE a pilot on an earth mission to Saturn. It is a mission that has gone badly wrong. A fire on board the Probe Mother Ship has wiped out all but a handful of the crew. You have managed to get the survivors into the one serviceable shuttle, which has taken three months to get you back to lunar orbit.

Your problem now is to land the shuttle on the Hover Pad at Moonbase Alpha. The three-month journey has left you with only 200 units of fuel. Your navigator has got you across millions of space miles to within 50 miles of the lunar surface. Now you find that the low level homing navcom has been damaged. The landing will have to be manual.

You are not sure of the landing tripod and have been advised to use the emergency hover pad at Alpha built in the early colonisation days. This pad holds the ship a few feet above the surface so enabling damaged ships to off-load passengers and cargo through extending umbilicals.

On reaching a predetermined height above the pad the ship is locked into position, held within a powerful magnetic field.

There is a moderate solar wind blowing your shuttle down-range from the pad towards the inhospitable moon peaks. If you should run out of fuel you have just one chance. If you can hit the eject button in time the survival capsule will blast away from the shuttle and you will be back to square one with another 200 units of the fuel to use.

If you are not quick enough or miss the eject button, well . . .

Program notes

This program uses the high level graphics to draw the lunar landscape and uses many of the other functions available on the Dragon.

Line 140 draws the shuttle at the foot of the screen.

Line 190 gets it and 300 puts it in position at the start of your descent.

Line 360 uses the *ppoint* function to detect a crash landing, whereas line 370 draws the magnetic field around the pad.

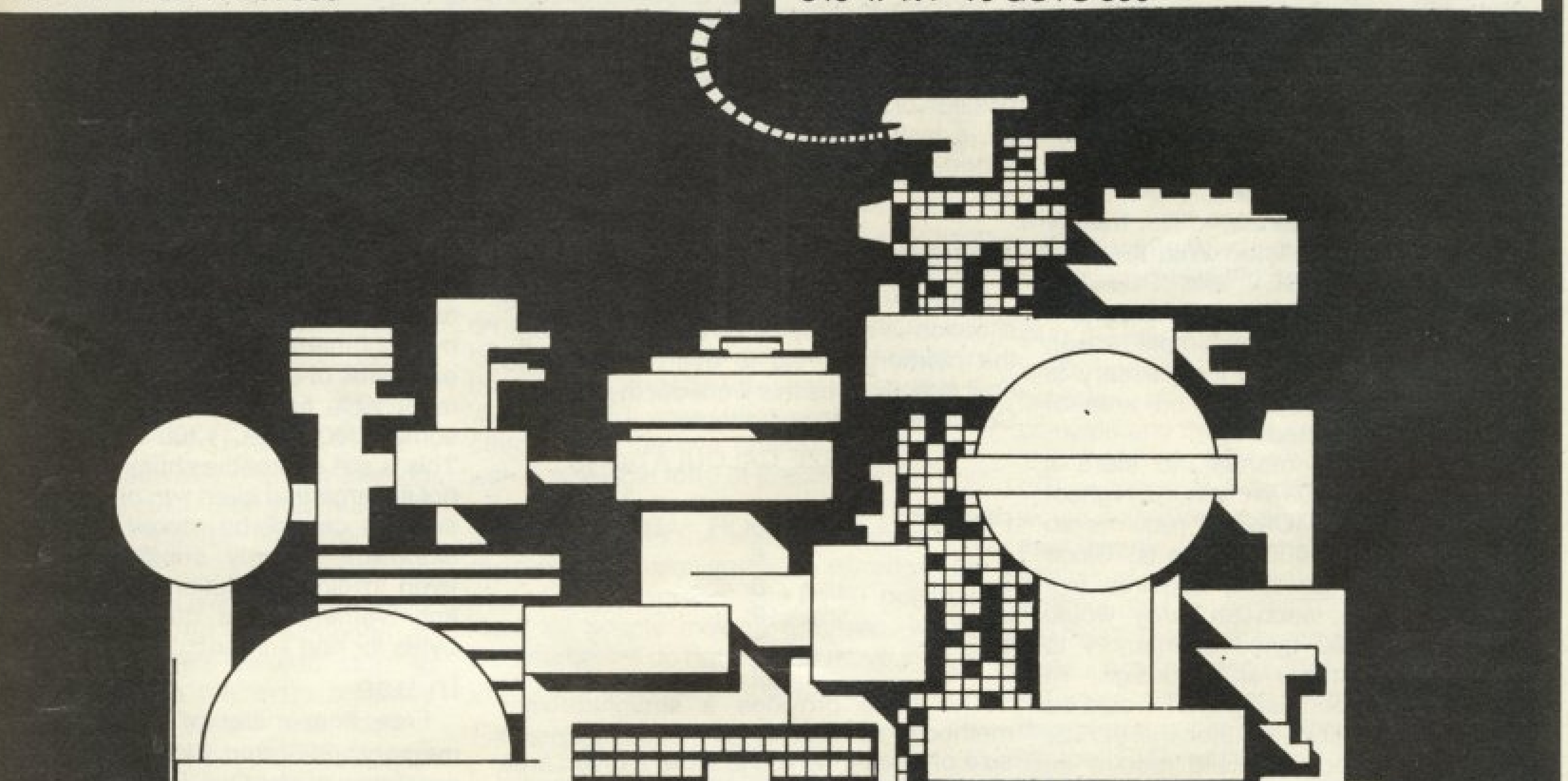
Line 500 disproves the manual when it says that the Dragon will only print text in black on green (or vice versa).

Moonbase Alpha was submitted by D. Windle of Nevern Road, Rayleigh in Essex.

```
10  "MOONBASE ALPHA COPYRIGHT D. E.
    WINDLE, 1982
20  CLS3:FOR C = 224 TO 255:T = 0
30  PRINT@C, CHR$(150):NEXT C
40  PRINT@256," MOONBASE ALPHA "
50  FOR D = 288 TO 319
60  PRINT@D, CHR$(150):NEXT D
70  PRINT@320," COPYRIGHT D. E. WINDLE
    1982"
80  P$ = "T255;05;CCDDEEGGAAFAABBAACC"
    :PLAY P$ + P$ + P$:T = 0
90  FOR T = 1 TO 2000:NEXT T
100 CLS:PRINT@163," YOU ARE THE PILOT OF A
    LUNAR SHUTTLE. YOU HAVE TO LAND YOUR
    SHIP ON THE PAD ON MOONBASE ALPHA.
    YOU HAVE ONLY 200 UNITS OF FUEL. GOOD
    LUCK. PRESS 'S' TO START DESCENT"
110 PRINT@385,"CONTROLS 'U' = UP 'R' =
    RIGHT"
120 S$ = INKEY$: IF S$ = "S" THEN 130 ELSE 120
130 PMODE 4,1;PCLS: SCREEN 1,1:COLOR
    0,5:DIM R(16,16,)
140 R$ = "BM110,177;U3E3R4F3D3G3BL
    4H3R4BR2R4U3L4BL2L4D3F3G3R2
    BL4R2E3R1BR2R1F3R2L4"
150 L$ = "BM0,170;U5R49F6R3F8R20
    U14E4R45F5R20E7R10F5R2F9R13D
    22R35U25E20"
160 B$ = "BM0,165;E7R2E9F12R2E8F
    4R2E6F4R2F4R2F12E15R3E5F5R2R1E4F6
    R3E8R2E7F12E9F12E4R2E7F6D3F16D2"
170 H$ = "BM206,147;R35D45L35U45"
180 DRAW R$: DRAW L$: DRAW B$
190 GET( 103,169) - (127,193),R,G
200 Y = 15:X = RND(200)
210 'FUEL SECTION
220 F = 232
```

Continued on page 29

```
230 DRAW" BM1,0;D9U4R2L2U5R4BR2D9R
    4U9BR2D9R4L4U4R3L3U5R
    4BR2D9R4"
240 LINE (32,2) – (F,4),PSET,BF
250 LINE (32,2) – (F,4),PRESET,BF
260 ' LEFT HAND DRIFT
270 W = RND(3)
280 X = X – W
290 Y = Y + 1
300 PUT (X,Y) – (X + 24,Y + 15),R,PSET
310 PAINT (0,180),0,0
320 PAINT (250,180)0,0
330 IF X < 3 THEN X = 5
340 IF X > 219 THEN X = 217
350 S = Y + 24
360 IF PPOINT(X,Y + 16) = 0 THEN 500
370 IF S = 184 THEN DRAW H$
380 IF S = 184 THEN 540
390 A$ = INKEY$
400 IF A$ = "U" THEN Y = Y – 2
410 IF X < 1 THEN X = 2
420 IF A$ = "U" GOSUB 480
430 IF A$ = "R" THEN X = X + 5
440 IF A$ = "R" GOSUB 480
450 IF A$ = "U" OR A$ = "R" THEN F = F – 2
460 IF F < 32 THEN 580
470 GOTO 230
480 PLAY"T 50;01;V5;CDEFGAB"
490 RETURN
500 CLS: PRINT@ 226," CRASH LANDING NO
    SURVIVORS": SCREEN 0, 1
510 PLAY "T200;05BGD; 04BGD;03BGD;
    02BGD;01BAGFEDC;T100;01BAGFEDC;
    T50;BAGFEDC;T10BGD"
520 CLS: PRINT@226," ANOTHER TRY? (Y/N)"
530 INPUT T$:IF T$ = "Y" THEN RUN ELSE 530
540 CLS: FOR N = 1 TO 1000: NEXT
550 PRINT@226," WELCOME TO MOONBASE
    ALPHA YOU HAVE "F" UNITS OF FUEL LEFT":
    SCREEN 0,1
560 PLAY "T6;04;CP4L2CL3D;03B;
    04CDEP4EFL2EL4DCL3DC;
    03L2B;04L2C"
570 RUN
580 CLS 8:PRINT@224," OUT OF FUEL . . . EJECT
    . . . EJECT . . . . . ."
590 N = 1
600 SOUND 225,1
610 N = N + 1
620 N$ = INKEY$: IF N$ = "E" THEN 630 ELSE 600
630 IF N < 10 THEN RUN
640 IF N > 10 GOTO 500
```



Getting the most out of graphics

***David Lawrence** introduces two of the most underused commands in Dragon Basic, GET and PUT, and shows how the manual got it wrong.*

LOOKING AT THE programs published in magazines suggests that one area which many Dragon users have not really begun to explore is the use of GET and PUT. If that's true, then it's a great shame because in many ways it's these two commands which make the Dragon's graphics capabilities stand out from those of other micros on the market.

Other micros have sophisticated line drawing commands, the ability to colour in limited areas of the screen, the ability to print the same design at a different angle. But how many have the ability to photograph part or all of their own screen for later use? Using GET and PUT to do just that, the painfully slow graphics procedures that so often bedevil Basic programs can be revolutionised.

So why is it, at the moment at least, that GET and PUT aren't more prominent in people's programs. Well, almost inevitably, a large part of the blame has to be laid at the door of the Dragon manual. Even those Dragon owners who have become used to detecting the inadequacies of the manual will probably not have discovered that when it comes to GET and PUT the manual is not merely inadequate it is a disaster. With the best will in the world, GET and PUT are consigned to the backwaters of programming by the manual for the simple reason that the amount of memory necessary to use the two commands on an area of screen is overestimated.

According to the manual, to store a screen area of 20*20 Pixels in the highest resolution mode (PMODE 4) requires an array of 20*20 or 400 elements. Since each element in an array requires five bytes of memory, such an array would require some 2000 bytes of memory to store only a relatively small design. To store a whole screen using GET would be a literal impossibility on either the 32K or 64K machine, since it would require an array of 192*256 elements, a total of 49152, taking up nearly a quarter of a million bytes of memory.

The largest design that a 32K machine could actually store, with no Basic program cluttering up the memory at all, would be some 6000 Pixels in size, say 80*75. What that means in practice is that anyone following the Dragon manual's advice on using GET and PUT would find themselves limited to the storage of no more than a few moderate sized designs before running out of memory. Far better, it would seem, to use relatively complex and considerably slower DRAW commands whenever any considerable area of screen is being delt with.

The GET command does not need anything like one whole array element to remember each separate Pixel on the screen. In fact, in the highest resolution mode it needs only 1 bit (eight bits to a byte, remember). Of course, when you come to think about it, this is quite logical. In PMODE4, with only two colours on the screen at any one time, all that needs to be remembered about the state of each Pixel position on the screen is whether it is on or off, a task for which a single bit is ideally suited since it, too, can only be either on or off.

Saving memory

In other, lower resolution, PMODEs the situation is slightly more complex, though the memory saving is even greater. So how exactly does this work out in practice? Consider the following table:

ARRAY SIZE CALCULATIONS

PMODE	DIVISOR	ARRAY ADJUSTMENT
4	8	5
3	8	5
2	16	5
1	16	5
0	32	5

The table provides a straightforward method of calculating the most economical size of array in which to store a given area of screen. It is used in the following way:

1) Determine the size, in high resolution Pixels, of the area of screen to be saved. Thus if the design to be saved covers the top left hand corner of the screen for 50 screen positions to the right and 50 screen positions down (i.e. 0-49 in both directions), its size will be 50*50 high resolution Pixels, or 2500.

The reason that high resolution Pixels are specified is that it does not matter at this state which PMODE you are in. All screen measurements on the Dragon are done on a 256*192 grid representing the Pixel positions in PMODE4. Your 50 by 50 area may actually contain 2500 useable positions (PMODE4) or 12*12 positions (PMODE0), according to the smallest block available in the PMODE you are using.

2) Divide the number you have obtained by the number in the divisor column opposite the PMODE you are using. If the result is not an integer number, then round it up to the next highest integer. In the case of our example, a 2500 Pixel area if we were using PMODE 4, 2500 would have to be divided by 8, giving a result of 312.5 or, when rounded up, 313.

3) You have now calculated the number of bytes you will need in order to store that area of screen — you must now calculate the minimum size of array that will contain this number of bytes. This is simply done by dividing by five. Once again, if the number is not an integer, it must be rounded up. Referring again to our example, 313 must now be divided by five, giving a result of 62.6, which rounds up to 63.

4) Having settled on the number of array elements you need, all that remains is to declare a one dimensional array equal to the number you obtained via the calculations above. In the case of our example this would require an array such as A(63).

The result of all this is that an area of screen which would have required, if the advice of the manual had been followed, an array of 50*50 elements, or some 7500 bytes of memory, has been reduced to 63 elements, or some 300 bytes. The requirement for a full screen is reduced to only some 6000 bytes (1200 array elements). This is still a sizeable chunk of memory but not so large that even two or three different screens cannot be stored, if necessary, alongside a fairly substantial program, even in high resolution modes. Quite an improvement on a quarter of a million bytes for one screen!

In use

Free from many of the constraints of memory, GET and PUT take on quite a new lease of life. One of the first areas you might like to explore is exactly the one which seemed so impossible before, and that is the storage of whole screens of graphics.

Imagine that you have written a game which involves a fairly complex screen design which needs to be reprinted afresh with some regularity. No doubt the original design was set up using some combination of DRAW and PSET commands. This is fine for the initial set-up but it can become extremely tedious if it has to be done too often.

The speedy solution, provided that you

```
100 PMODE 4:PCLS:SCREEN 1,0
110 FOR I=1 TO 191 STEP 2:FOR J=1 TO 255 STEP 2:PSET (J,I):NEXT J,I
120 DIM A(1229)
130 GET (0,0)-(255,191),A,G'SLIGHT PAUSE HERE
140 PCLS
150 IF INKEY$="" THEN 150'PRESS KEY TO REINSTATE SCREEN
160 PUT (0,0)-(255,191),A,PSET
170 GOTO 170
```

Listing 1 – a whole screen can be stored in an array

```
170 IF INKEY$="" THEN 170'PRESS KEY WHEN READY TO SAVE
180 CSAVEM "PICTURE",1536,1535+6144,6144
```

Listing 2 – saves your graphics as a block of memory on tape

have around 6000 bytes of memory left, is to set up the initial screen and then GET it into an array. From then on, whenever you need to reset the screen you need only PUT the array back on to the screen. While for a simple design this is hardly worth it, the difference in speed for a complex design is dramatic and can make the difference between an amateurish program and one with all the slickness of a professional product. *Listing 1* shows how a whole screen can be stored in an array and later replaced on the screen.

On occasions, however, you may well be pushed for space. The lines necessary to set up the design in the first place may take up a considerable amount of memory and it may not be practical to add to that a large array, no matter how much you desire to speed up the program. In that case, why not consider setting up the initial design, storing it on tape separately from the program itself, then reloading it when the same is played and GETting it for use in the game — doing away with the need for all those DRAW,PSET and PRINT commands in the main program itself.

Procedure

All that you need to do is to design your screenful of graphics. When it is done to your satisfaction, save it as a block of memory on tape using lines such as those in *Listing 2*, which are meant to be added to *Listing 1*.

The three numbers in line 180 refer to the beginning of the memory area, the finish point and the number of memory locations. Provided that you are always starting at page one in the video memory, the figures for the various PMODEs would be as follows:

PMODE	START	FINISH	SIZE
0	1536	3071	1536
1	1536	4607	3072
2	1536	4607	3072
3	1536	7679	6144
4	1536	7679	6144

If you wish to use video memory starting at any other page but 1, then you must add 1536*(page no.-1) to the first two numbers.

You can now add *Listing 3* to the main program. It will pick up the screenful of graphics from tape (it's best to store it immediately following the main program on the tape so that there's less fiddling about). All the graphics commands can thus be omitted from the main program. Not only does this technique allow you to save memory, it's also extremely impressive when you show the program to your friends, with complex graphics appearing out of nowhere in a program which apparently has no graphics commands!

You will note that in these listings I have included the 'G' option in the GET statements, which means simply that full graphic detail is picked up into the relevant array. This appears to make no difference to the memory demands. There is no apparent hard and fast rule for when 'G' must be included, it is simply that some-

```
190 IF INKEY$="" THEN 190'PRESS KEY WHEN READY TO LOAD AGAIN
200 CLEAR:PCLS:DIM A(1229)
210 AUDIO ON
220 CLOADM "PICTURE",0
230 PMODE4:SCREEN 1,0
240 GET (0,0)-(255,191),A,G
250 PCLS
260 IF INKEY$="" THEN 260'PRESS KEY WHEN READY TO REINSTATE
270 PUT (0,0)-(255,191),A,PSET
280 GOTO 280
```

Listing 3 – picks up the screenful of graphics from tape

times its omission can give some strange results — whether you can leave it out is really a matter of trial and error in your particular application.

You can also omit the PSET option from the PUT statement and obtain a big increase in speed. PSET, PRESET, OR, AND and NOT (discussed below) can all be used in a put statement without the inclusion of 'G' in the GET statement. Test this for yourself by removing the 'G' from line 130 of *Listing 1* and running the program. Now remove the PSET from line 160 and notice the increase in speed.

GET and PUT are not limited in their usefulness to applications involving the whole screen, or even large sections of it. Properly used they represent not only a way to higher speed but a massive increase in the overall graphics capabilities of the Dragon compared to a machine which only has such commands as PSET and DRAW. Many users have not discovered this because they have yet to see the real relevance of the PUT options, PSET, PRESET, AND, OR and NOT.

The simplest of these options are PSET and PRESET. To include PSET or PRESET in the PUT statement (e.g. PUT (X1, Y1) – (X2, Y2),A,PSET [or PRESET]) simply means that whatever was taken from the screen will be replaced as it was found (PSET) or inverse (PRESET).

Anything which was originally in the area on to which the design is being placed will be obliterated by the design contained in the array in exactly the same way that it would if you were over-printing text. In effect, what the two options provide you with is a flexible form of graphics PRINT @ statement — flexible because you can specify down to the last Pixel where the character or design you have stored will be placed on the screen. The PSET option is ideal for simple moving graphics, where the characters do not have to move in front of existing material on the screen.

Smooth moving

When the character to be used has been defined (probably using DRAW) simply GET a screen area which contains the character and a margin of blank screen on each side. If you intend to move the character only one Pixel at a time then a one Pixel blank margin will do. The character can now be moved smoothly around the screen, always obliterating its previous position as it moves.

Listing 4 gives a very crude example of a circle which moves in a straight line across the screen but there is no difference in principle for more complex movement. Changing the PSET in line 160 to PRESET will illustrate the difference between the two options.

The remaining three options available for PUT really do need to be explained together since it is not so much their individual capabilities but their interactions which make them so interesting:

a)OR This option makes your PUT statement perform in a way very similar to the OVER characteristic on the Sinclair Spectrum. What it does is to superimpose the design stored in the array on to whatever is currently in the target area of the screen. Thus the original screen contents are not obliterated unless the design in the array covers all the pixels that were on (or rather 'set') before it was placed there.

The result is a mix of both what was on the screen and what was in the array, with any Pixel in the designated screen area being set if EITHER it was set before the PUT statement was executed OR it was set in the design stored in the array. As we shall see, this can be extremely useful when it is desired to move something through or over what is already on the screen.

b)AND This leaves only those Pixels set which were originally set BOTH in whatever was on the screen before the PUT statement was executed AND are in the design stored in the array. If you were to PUT a character on the screen and then PUT it again in the same position using the AND option, you would see no effect because all the Pixels which are set coincide and thus they all remain set.

If, however, you were to put a small outline circle within a larger one, they would both disappear completely because they share no common Pixels. Usually AND is at its most useful when employed with the next option, NOT.

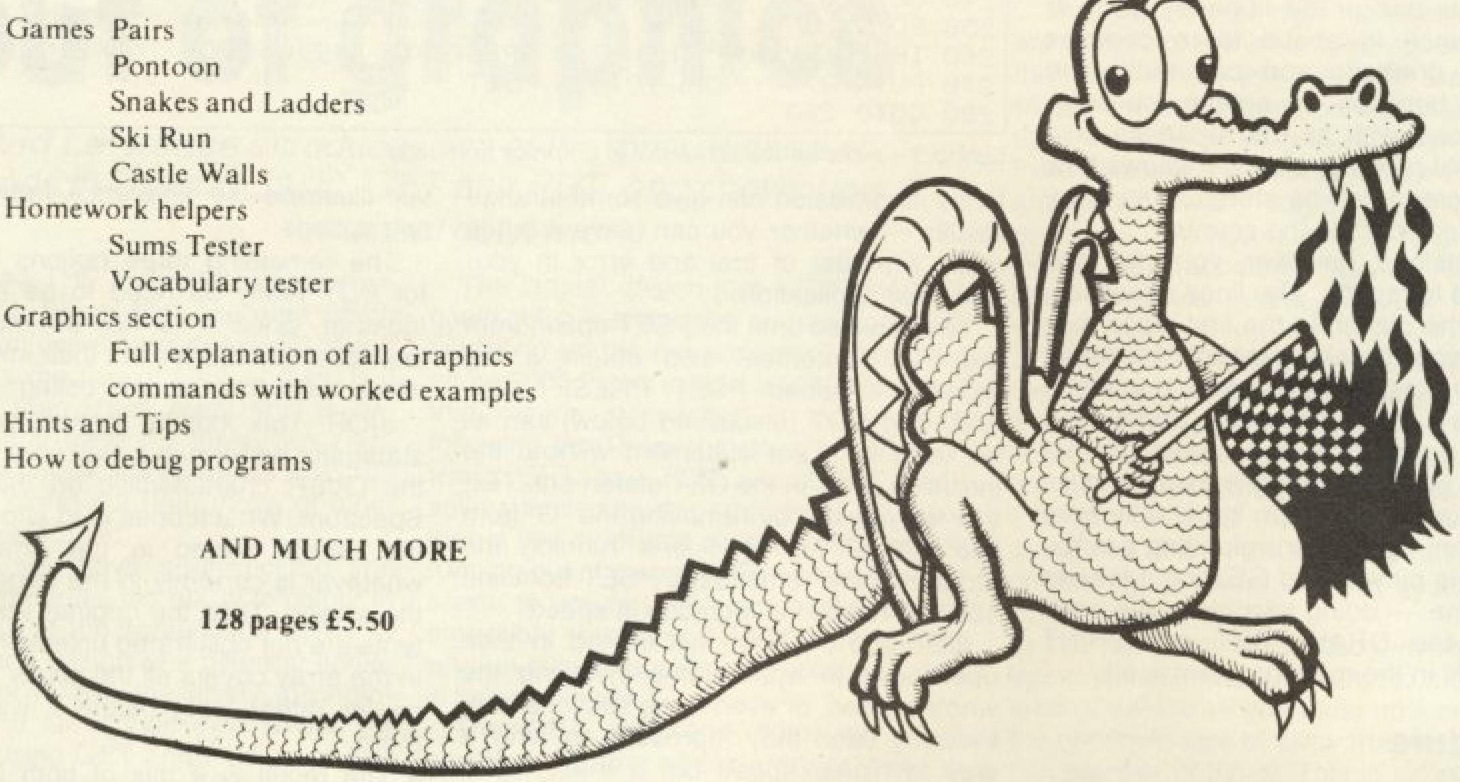
c)NOT This may seem a strange one at first, since provided that the array is of sufficient size to cover the screen area designated, what is in the array specified in the PUT statement makes not the slightest difference to what happens on the screen. What does happen is that within the specified rectangle on the screen any Pixel that was previously set is reset (switched off) and any Pixel that was previously reset is set. The contents of the rectangle are thus inverted. At first sight this might not seem incredibly inspiring but, together with AND it enables us to perform a very selective erasure from the screen.

Listing 5 illustrates how an erasure of one design can be achieved without necessarily blotting out another design with which it shares an area of screen. ▶

LOAD AND GO WITH YOUR DRAGON

32 PROGRAMS FOR YOUR DRAGON 32

- Games Pairs
  - Pontoon
  - Snakes and Ladders
  - Ski Run
  - Castle Walls
- Homework helpers
  - Sums Tester
  - Vocabulary tester
- Graphics section
  - Full explanation of all Graphics commands with worked examples
- Hints and Tips
- How to debug programs



AND MUCH MORE

128 pages £5.50

PHIPPS ASSOCIATES

Dept D FREEPOST EM463 (No stamp required)
99 East St, Epsom, Surrey KT17 1BR.
Telephone 03727-21215. 24hr phone service.

Prices include postage but for air mail delivery in Europe add 90p (outside Europe add £1.70) per item.

Access and Barclaycard accepted

DRAGON SOFTWARE

We offer a new mail order software service and can supply programs from most of the leading software houses. The following are a few examples — if you can't see what you want, please ask.

MICRODEAL	DRAGON DATA
Donkey King*	Astrobust £19.95
Katerpillar Attack	Black Sanctum £7.95
Cosmic Zap	Chess £24.95
Talking Pararaid Attack	Galae Attax £19.95
Backgammon	Ghost Attack £23.50
Galactic Ambush	Berserk £19.95
These and all other Microdeal	Meteoroids £19.95
games available £7.95	Cosmic Invaders £19.95
*with Free Badge!	

QUARX DATA	SALAMANDER	MC LOTHLORIAN
Flight £6.50	Dragon Trek £9.95	Samurai Warrior
M.R.C. £6.50	Wizard War £7.95	Tyrant of Athens
Pontoon £4.95	Golf £7.95	Warlord
Look & Learn £4.95	Games	£5.90 each
Home Plan £7.95	Compendium £7.95	
	Vulcan	
	Noughts & Crosses £7.95	

A & F SOFTWARE	JOYSTICKS	PRINTER
Deadwood £6.90	£19.95 pair	GP100A £215

WORD PROCESSING

JUNIPER: Menu/cursor + driver. Menu options include CREATE/EDIT/SAVE/VERIFY/PRINT/TAPE-PRINT. Edit functions include repeat key, tabs, cursor up, down, left, right, delete up/down (lines), right, left (character) will operate with any printer. **£25.00**

MICRODEAL Telewriter similar to Juniper but with 51 x 24 display, block copy move/delete. **£49.95**

ALL OUR PRICES INCLUDE VAT & P&P.
SEND CHEQUE/PO/or ACCESS NO. TO

JUNIPER COMPUTING

8 PEMBROKE GREEN, LEA, MALMESBURY, WILTS SN16 9PB
TEL: 06662-2689

BOND SYSTEMS

YOUR DRAGON INCREASES
YOUR LEARNING CAPABILITIES

VOCAB FRENCH and **VOCAB GERMAN.** Transfer the 700 most commonly used words from computer memory into your memory.

TOUCH TYPING. Your DRAGON can type at up to 100 words/minute. Can you? New system for quickly acquiring accurate high speeds. Typing speeds indicated throughout each session. Ideal for teaching yourself or others a useful skill.

TRAINER. An almost universal self-teaching guide to learning almost everything. If your tape recorder uses the REM or AUX plug, use this tape. Examples of MATHS, GEOGRAPHY, HISTORY, SPELLING, and ADVANCED SPELLING are shown, together with details for the introduction of your own data.

WAFFLE DODGER. When you write do you use plain English or is it complicated and not understood? This program shows how good/bad your prose is.

VOCAB FRENCH...... **£5.00**
VOCAB GERMAN..... **£5.00**
TOUCH TYPING **£5.00**
TRAINER.............. **£5.00**
WAFFLE DODGER.... **£3.50**

BOND SYSTEMS
15 BELMONT ROAD
HARROGATE
NORTH YORKSHIRE
HG2 0LR

```
100 DIM A(2)
110 PMODE4,1:PCLS:SCREEN 1,0
120 LET D$="BM100,100;U2;E2;R2;F2;D2;G2;L2;H2":DRAW D$
130 GET (99,95)-(107,103),A,G
140 PCLS
150 FOR I=10 TO 100
160 PUT (I,I)-(I+8,I+8),A,PSET
170 IF INKEY$="" THEN 170'THIS LINE IS OPTIONAL
180 NEXT I
190 GOTO 190
```

Listing 4 – shows a circle which moves in a straight line across the screen

```
100 DIM A(2),B(2)
110 PMODE4,1:PCLS:SCREEN 1,0
120 LET D$="BM100,100;U2;E2;R2;F2;D2;G2;L2;H2":DRAW D$
125 IF INKEY$="" THEN 125'LINES ENDING IN 5 ARE OPTIONAL
130 GET (99,95)-(107,103),A,G
140 PUT (99,95)-(107,103),A,NOT
145 IF INKEY$="" THEN 145
150 GET (99,95)-(107,103),B,G
160 PCLS
170 PUT (100,100)-(108,108),A,PSET
175 IF INKEY$="" THEN 175
180 PUT (107,100)-(115,108),A,OR
185 IF INKEY$="" THEN 185
190 PUT (100,100)-(108,108),B,AND
200 GOTO200
```

Listing 5 – how to erase one design without blotting out another

```
100 DIM A(1),B(35),C(1)
110 PMODE 4,1:PCLS:SCREEN 1,0
120 DRAW"BM100,100;U2;E2;R2;F2;D2;G2;L2;H2"
130 GET (100,96)-(106,102),A,G'NB NO BLANK BORDER
140 PCLS
150 FOR I=10 TO 200 STEP 10
160 PUT (I,100)-(I+6,106),A,PSET
170 NEXT I
180 GET (10,100)-(206,106),B,G
190 PCLS
200 DRAW "BM100,100;E3;BL3;F3"
210 GET (100,97)-(103,100),C,G
220 PCLS
230 FOR I=15 TO 200
240 PUT (I,101)-(I+3,104),C,OR
250 FOR J=1 TO 50:NEXTJ'TIMING LOOP TO SLOW IT DOWN
260 PUT (10,100)-(206,106),B,PSET
270 NEXT I
280 GOTO 280
```

Listing 6 – provides a small cross moving along a line of circles

```
100 DIM A(1),B(35),C(1),D(1)
110 PMODE 4,1:PCLS:SCREEN 1,0
120 DRAW"BM100,100;U2;E2;R2;F2;D2;G2;L2;H2"
130 GET (100,96)-(106,102),A,G'NB NO BLANK BORDER
140 PCLS
150 FOR I=10 TO 200 STEP 10
160 PUT (I,100)-(I+6,106),A,PSET
170 NEXT I
180 GET (10,100)-(206,106),B,G
190 PCLS
200 DRAW "BM100,100;E3;BL3;F3"
210 GET (100,97)-(103,100),C,G
220 PCLS
225 PUT (10,100)-(206,106),B,PSET
230 FOR I=15 TO 200
235 GET (I,101)-(I+3,104),D,G
240 PUT (I,101)-(I+3,104),C,OR
250 FOR J=1 TO 50:NEXTJ'TIMING LOOP TO SLOW IT DOWN
260 PUT (I,101)-(I+3,104),D,PSET
270 NEXT I
280 GOTO 280
```

Listing 7 – gives faster and smoother movement for the cross

◀ If you enter and run the listing you find that two circles are printed so close to each other that the blank borders of the rectangles containing them actually overlap. They do not erase each other because the second circle is printed with the OR option.

One of the two circles is erased when any key is pressed by the printing over it, with the AND option, of an inverse copy, which was created in line 140 by using PUT and NOT and then copied into a second array. When the inverse copy is placed over the original using AND, there are no set Pixels corresponding, by definition, so the circle is obliterated.

However, all the blank areas in the original design are replaced by set Pixels in the inverted array, so where there is a set Pixel on the screen which is not a part of the design being erased, it coincides with a set Pixel in the second (inverted) array and true to the rules of AND it remains set. Unfortunately if there are any Pixels shared by the two designs, for instance if the two circles had been interlocking, then those Pixels are erased.

More complex

The problem of erasing designs where they actually overlap others is not always quite as straightforward. Sometimes, it is true, the number of Pixels shared by the two designs is so small that their disappearance will make very little difference but this is not always the case. It is then necessary to reprint the design that you wish to leave on the screen.

Listing 6 shows how this technique can be employed to provide a small cross moving along a line of circles. All that happens here is that the cross is PUT on to the screen with the OR option, so that it appears over any circle in that position. To erase the cross before it moves to the next position, all that is necessary is to PUT the whole line of circles on to the screen again using PSET, which erases the cross. The overall effect is one of motion by the cross along the line of circles and the method can be easily adapted to a variety of applications, especially in games.

Where the design over which you wish to move something is changing, so that you cannot keep PUTting the same background on to the screen, the solution is to use GET to photograph the area of screen on to which the moving design is about to be placed and restore the background to that state when the moving design has to be erased.

Listing 7 illustrates how this can be done with the circles and the moving cross and shows that the movement is both faster and smoother than the previous method mentioned.

No doubt on first reading this seems to be overly complicated, but half an hour's playing with the listings provided will reassure you that GET and PUT hold no horrors. What they do hold is a great deal of promise that many of us have only just begun to realise. More than that, they provide a timely reminder that while few modern micros do less than their manuals claim, the good ones are often capable of a lot more.

Getting to grips with handling file procedures

***David Lawrence** introduces some advantages and pitfalls of learning to use the Dragon's data files.*

ALMOST ANY serious programming on the Dragon will sooner or later require the use of data files; that is the saving of bodies of data on tape so that they can be used later, thus avoiding the need to manually re-enter information each time a program is reloaded.

Data files are also invaluable in developing programs. Since the Dragon unfortunately clears its variables every time a change is made to a program line, a routine to save and reload the variables can be a life saver when the program is being debugged.

At the same time, data files can be one of the most frustrating aspects of Dragon programming for the simple reason that when things go wrong you cannot see what is on the tape to identify the problem. The purpose of this article is to identify some of the major pitfalls in using data files and to suggest ways of avoiding them. In order to illustrate the methods employed, a specimen data file handling module is included.

One important point before starting to work with data files concerns the control of the cassette recorder through the remote socket.

A positive pain

No matter what the benefits when it comes to saving space on a tape, the Dragon's control of its own cassette recorder can be a positive pain when it comes to loading or saving data files while a program is running, since the user is no longer able to switch the motor on and off at will to position the tape correctly, without disturbing the program in its operation.

One solution to the problem is, of course, to leave the control line unplugged. While this is perfectly acceptable when it comes to the loading and saving of programs or during the saving of data it can cause problems when done during the loading of data files from tape. The reason for this is that in loading data from tape the Dragon will regularly pause to ensure that the data so far accepted is properly placed in the memory before accepting another batch. If the control line is connected then the cassette motor will be switched off during this pause but without the control line the motor continues running and the result can be that some of the data on the tape is missed.

The real solution to the problem is to include in the data file module a routine such as that shown in *listing one* — it is designed to allow easy positioning of the tape, with the motor on and the contents of the tape relayed through the TV speaker,

```
1000 AUDIO ON:MOTOR ON:PRINT:INPUT "POSI
TION TAPE THEN PRESS enter  (MOTOR IS ON
)";Q$:MOTOR OFF
1010 PRINT:INPUT "PLACE RECORDER IN CORR
ECT MODE  THEN PRESS enter";Q$
1020 PRINT:PRINT "FUNCTIONS AVAILABLE:",
"1)SAVE DATA",,"2)LOAD DATA":INPUT "WHIC
H DO YOU REQUIRE:";Q:ON Q GOTO 1040,1100
1030 RETURN
```

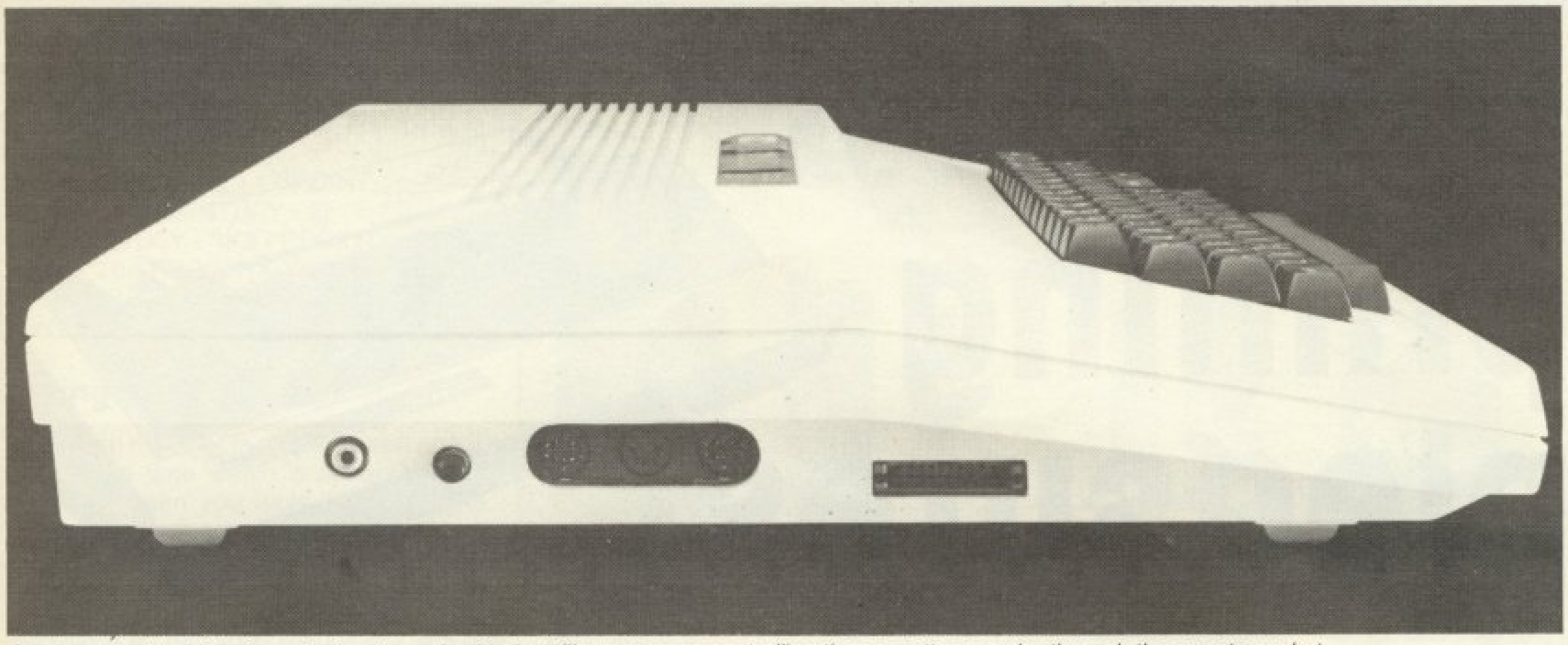
Listing 1 – designed to allow easy positioning of the tape with the motor on (the £ symbol should be read as #)

```
1040 MOTOR ON:FOR I=1 TO 10000:NEXT I
1050 OPEN"O",£-1,"DATAFILE"
1060 PRINT£-1,N1,N2,V1,V2,V3,V4,V1$,V2$
1070 FOR I=0 TO N1-1:PRINT£-1,A$(I):NEXT
1080 FORI=0 TO N2:PRINT£-1,A(I):NEXT
1090 CLOSE£-1:RETURN
```

Listing 2 – printing a header for tape handling (the £ sign should be read as #)

```
1100 OPEN"I",£-1,"DATAFILE"
1110 INPUT£-1,N1,N2,V1,V2,V3,V4,V1$,V2$
1120 FOR I=0 TO N1-1:INPUT£-1,A$(I):NEXT
1130 FOR I=0 TO N2:INPUT£-1,A(I):NEXT
1140 CLOSE£-1:RETURN
```

Listing 3 – combining the listings gives a module adaptable for most filing purposes (the £ symbol should be read as #)



One important point before starting to work with data files concerns controlling the cassette recorder through the remote socket

before the user hands back control of the cassette recorder to the Dragon for either loading or saving.

Before saving data there is another important point with regard to tape handling which is illustrated by line 1040 of *listing two* — the printing of a header. If you have experimented with storing data on tape then you have probably already discovered that the Dragon is extremely finnicky about what is on the tape when it first begins to listen.

If there is irrelevant noise or the improperly erased remnants of some previous file immediately before the specified set of data, an error is flagged and the program stops. For this reason it is vital to ensure that a clear space is placed before each data file on the tape. The length of the header (as dictated by the size of the loop) is a matter of preference but having one as long as that shown has the advantage that if you accidentally begin recording at the very beginning of the tape it will carry you safely over the tape leader and no data will be lost.

Having set up your tape correctly, the next necessity is to open communication between the Dragon and the cassette recorder (which the Dragon knows under the name of device −1 or #−1 for short). This is done automatically when programs are saved and loaded but with data you have to do it yourself.

The file you open can be one of two types, an output file (line 1050) for saving data to tape or an input file (line 1100) for loading data from tape. The file you wish to open must also have a name. If you are saving data then it will be saved under that name whereas if you are loading data the Dragon will not load a file with a different name.

So what can be saved? In fact anything that can be stored in the Dragon's memory, though some of it requires some trickery. Examination of the specimen module should show that any numeric variable (or indeed a literal number if you like) and any text string can be saved or loaded simply by name, using the format PRINT#−1 or INPUT#−1. The main limitation is that you cannot save non-standard characters, such as the low resolution graphics characters.

```
100 PRINT#−1,LEN(A$):FOR I=1 TO
LEN(A$):PRINT#−1,ASC(MID$(A$,I,1)):
NEXT 200 INPUT# − 1, L:LET A$ = "":FOR
I=1 TO L:READ#−1,CH:LET A$=A$
+CHR$(CH):NEXT
```

Table 1 – how to save a string of graphics characters

Should you wish to save a string of graphics characters — say a string that prints a design — you must translate each character into its ASCII code, save that number and then, on reloading, translate each number back into a character. The two specimen lines in *table one* show how it is done.

Loading

Another point of interest is what is not present in *listing three*, the loading routine. According to the Dragon manual it is almost obligatory to make use here of the EOF function. This stands simply for End Of File and is a way of detecting the special marker which denotes the end of the data stored on tape. The importance of this is that if the Dragon attempts to read beyond the end of the file it will encounter unintended data and probably stop with an error. EOF can be used to avoid this by including a line such as:

100 IF EOF(−1) THEN GOTO 200

just before each group of items is read from the tape. If the end of file marker is encountered, the program execution can be made to jump out of the module which reads the data. This function can clearly be useful in certain circumstances but in most cases it is unnecessary and can hide programming errors. The reason that it is unnecessary is that for the vast majority of programs, the number of items of data of any particular type is, or should be, constantly recorded and updated in the form of variables used by the program.

In the module shown, apart from a few loose variables, the program is storing two sets of items on tape (the two arrays A$ and A) and the number of items in each set is recorded by the variables N1 and N2. All that is needed is to save N1 and N2 first and, when the data is picked up from tape, the module will know exactly how many items to read in order to input the whole file.

The reason I say that this is a better way of saving and loading data is that it will automatically highlight any errors in the way that data is being stored. Either precisely the right amount of data is present or an error is likely to occur. Using EOF this can be disguised since, whether or not the correct amount of data is present, the loading will terminate successfully at the end of the file.

It is as well to remember that you cannot load data into arrays that haven't been created. That is to say, it's no use loading your program from tape and then calling up the data file module unless, either in the main body of the program or in the routine represented by *listing three*, you recreate with DIM statements all the arrays that once held the data but which were not saved on tape. It's easily forgotten.

Lastly, when loading or saving is complete, the file you have opened must be CLOSEd. The reason for this is that the Dragon can only cope with one file at a time, and failure to close the previous one will result in an error being flagged the next time you try to communicate with the cassette recorder.

The three specimen listings, combined into one module, are easily adaptable for most filing purposes, so without more ado you can get down to the task of applying your Dragon to some serious file handling.

Dragon's printing pleasures

***Keith Brain** explains how to make the most of your printer, from control codes to screen copying*



THE SEIKOSHA GP-100A is a versatile and popular low-cost dot-matrix printer with the Dragon-type Centronics parallel interface. It can produce both single and double width character and also has a dot-addressable graphic printing mode.

Printing mode selection

On power up single-width characters are produced but normal characters, double width characters or graphics can be selected by simply sending the appropriate control code (CHR$(15), CHR$(14) and CHR$(8), respectively) to the printer by typing PRINT#−2,CHR$(X). The new mode will be held until it is countermanded or the printer turned off. If you find no characters print out you have probably left it in graphics mode.

Character sets

The GP-100A has four alternative character sets available and you need to decide which one to use. The selection is made by means of an internal DIP switch, so this is really a set-and-forget decision. Most of the characters in each set are the same but twelve differ. Two of the sets produce unusual German and Swedish accented symbols, and so are of little value to the average user. The other choice is between the UK and USA sets, which only differ in that in the UK set the hash sign is replaced by the sign for the pound sterling. Certain other useful characters such as various brackets, arrows and Greek symbols are also available via the appropriate character code.

Listing Basic programs

The first use of a printer is to make program listings. Typing LLIST will print out the current program until the program end is reached, or BREAK is pressed. The full width of the paper is used (80 columns) and lines automatically wrap round. If you want larger print out change to double-width characters before typing the command LLIST.

The PRINT#−2, command outputs ASCII character codes to the printer and the effects of punctuation are similar to on-screen. If you use this command in direct mode then nothing will print until you press ENTER. If you want to include blank lines on the printer just send the carriage return control code CHR$(13).

Defaulting

The default print start setting is at the left margin of the paper. If you want to move this to the right you need to first indicate that you want to make a change of print position (by sending CHR$(16) (POS)), and then specify how many character units you want to move with the next two bytes. This data must be repeated at the start of each print line so it is best dealt with in a



FOR/NEXT loop. Some print formatting can also be achieved by the PRINT#−2, USING format; output list command.

The text screen memory runs from 1024 to 1535 and you can easily PEEK what is on the screen. However, if you try to copy the characters on the text screen to the printer by PRINT#−2, CHR$(PEEK(1024+N)) you may not get the result you expect, as not all the characters are stored in the screen memory of the Dragon as their ASCII codes.

To convert screen PEEKS to ASCII codes PEEKS between 96 and 126 need to have 64 subtracted, PEEKS between 1 and 26 need 96 adding to them, 0 needs to be changed to 32, and PEEKS between 27 and 95 print correctly.

That is not as bad as it sounds at first and a whole screen copy can be produced rapidly by these two lines.

```
10 FORY=0TO15:FORX=0TO31:
A=PEEK(1024+X+(Y*32)):IFA>95
ANDA<127THENA=A−64ELSEIFA>0
ANDA<27THENA=A+96ELSEIFA=0
THENA=32
20 PRINT#−2,CHR$(A);:NEXTX:
PRINT#−2,CHR$(13);:NEXTY
```

A copy routine for the hi-resolution screen is invaluable but is a bit more complicated to achieve. The Dragon hi-resolution screen is mapped differently according to the PMODE selected but fortunately the coordinate specifications used are the same for all modes.

To see how the display is produced set PMODE0,1:Screen1,0 and then CLS when you will see a plain screen. If you now POKE 1536,255 a dark band will appear at the extreme top left of the screen. If you try poking smaller numbers into 1536 you will find this band changes and breaks up. Poking 1537 will similarly affect the narrow strip which is to the right of 1536.

The Dragon hi-resolution mapping system thus sets each screen point as one bit in this mode, moving from left to right, but with the most significant bit on the left. If you PSET(0,0,0) you will turn on the point in the extreme top left corner of the screen (that is the same as POKE 1536,128).

The GP100A looks at printing graphics in an up-and-down fashion, rather than the side to side set-up of the bits in the screen memory. In graphics mode seven dots are addressed by setting bits 1 to 7 in a single byte, with bit 8 always set. To convert the screen image to printout we therefore need to look at screen position 0,0. If that bit is set then set bit 1 of the first byte to be sent to the printer.

Moving down

We now need to move down the Y axis one point to 0,1. If this bit is set then set bit 2 of the first printer byte. This must be repeated until seven bits have been tested, and then the eighth bit set to complete the first byte.

Fortunately Dragon Basic has PPOINT which actually checks the status of each bit on the screen (using the same coordin-

ates for all modes) and returns a 1 or 0, but we still need to set the printer bits by adding the appropriate numbers to the first byte. The sequence for the first byte is therefore:

```
10 A=PPOINT(X,Y)*1+PPOINT
(X,Y+1)*2+PPOINT(X,Y+2)
*4+PPOINT(X,Y+3)*8+PPOINT
(X,Y+4)*16+PPOINT(X,Y+5)
*32+PPOINT(X,Y+6)*64+128:
PRINT#-2,CHR$(A)
```

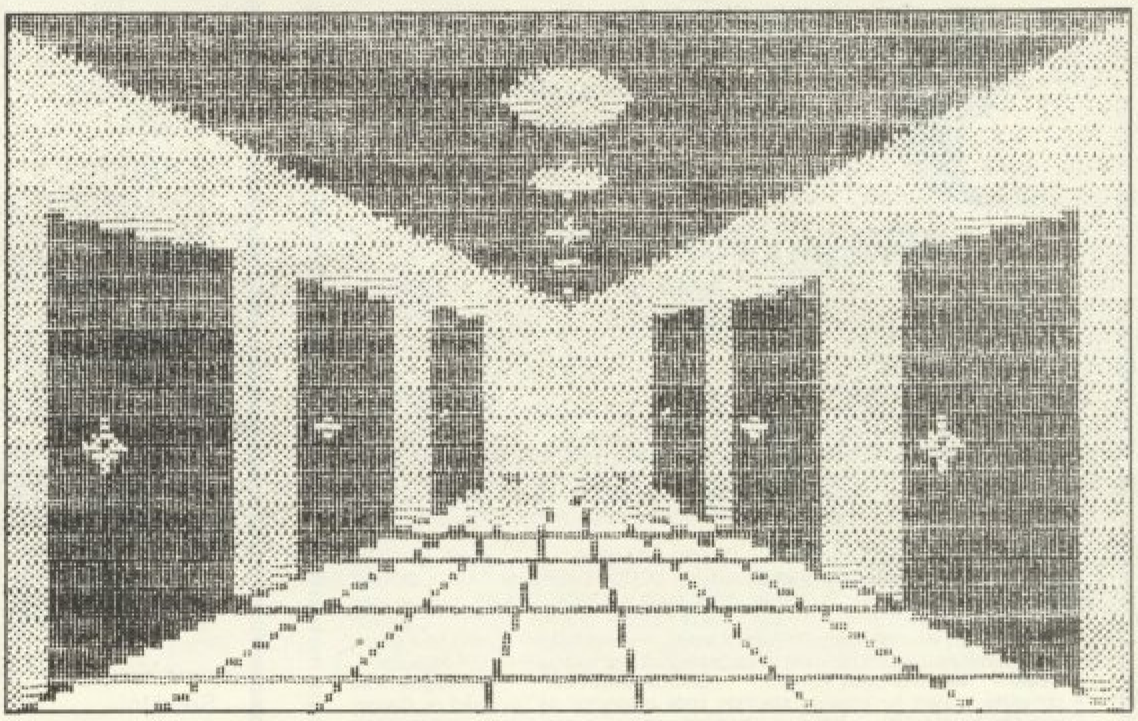
Moving across

Thus you move across the screen one X column at a time which is seven Y rows deep, and when you have collected 255 bytes you need send a carriage return CHR$(13), and then move down the Y axis of the screen seven rows and start to calculate the next line of graphic print. Note that in graphics mode no gaps are left between any of the lines so they will join up neatly.

In the four-colour modes bits are set in pairs to indicate the four colours:

	First bit	Second bit
First colour	OFF	OFF
Second colour	OFF	ON
Third colour	ON	OFF
Fourth colour	ON	ON

and a consequence of this is that a four colour screen will print out as white, black, and left and right-handed zebra stripes. There are ways around even that problem which allow you to produce more subtle shading, as well as scaling and partial

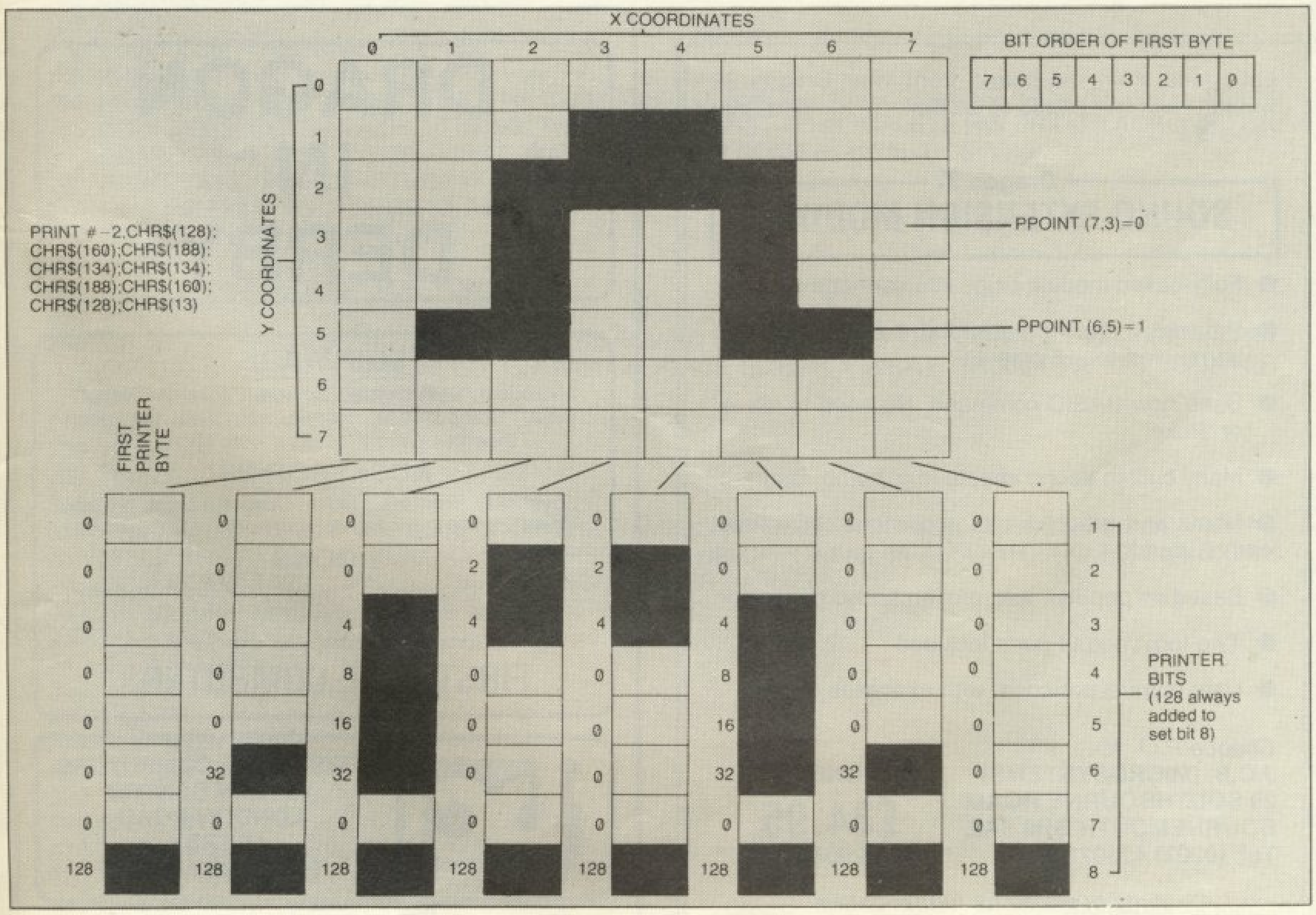


prints, as shown in some of the examples.

It is not necessary to have the screen-print routine included in a program producing graphics you may want to print. You can store graphics (or even text) pages on tape by using CSAVEM, and appropriate memory values according to the PMODE, and then CLOADM them back for later printing.

Although the screen to print conversion through Basic is not exactly superfast it proceeds at a reasonable rate especially considering there are up to 48K calculations to be made. You can speed it up with POKE&HFFD7,0 but be sure to remember that you must reset the speed before you can save or load.

A comprehensive menu-driven program for printing the Dragon hi-resolution screen with the Seikosha GP100A is available from the author. The program allows full or partial screen printing in black and white, inverse black and white, and four shades. It also includes a scaling-up facility, and details of how to produce custom shading patterns. It costs £5.95 on cassette or £2.95 as a listing. Keith Brain's address is Penygaer, Groeswen, Cardiff CF4 7UT.



Conversion of a high resolution screen to a GP1000A graphics format

Send us your Dragon programs, beginning with a general description and explaining how the program is constructed. Take care that the listings are bug-free. We pay £6 for each program published.

Open File

Hex

This program shows the graphics capability of the Dragon and the Tandy colour micros. Written in mode 3, a colourful and impressive display can be achieved with the minimum of programming. The game itself is a new version of an old Chinese boardgame.

The instructions are included in the print-out, but the object of this two-player game is to be the first to build a "bridge" of counters across the hexagonal board, whilst trying to block your opponent by trapping him/her. The first to do so wins the game (after a game is won the program must "break" and run again). This program was submitted by D. Owen of Prestwich, Manchester.

Program notes

Lines	
60-130	Opening program by asking for instructions.
140-350	Instructions.
370-400	Sets up mode, colour of screen and characters.
410-1010	Draws screen layout.
1040 & 90	Test to determine set colour of point (X,Y).
1160	As there are 11 rows on the board, CHR$(58), ":" is used for 11.
1170	As above, "0" is used for row 10.
1200-1300	The INKEY$ function finds letter on the keyboard being selected and converts it into a row number.
1320	Computes area being tested in lines 1040 and 1090.

Keys	
A-K	Rows 1-11 (rows A-K on screen).
1-:	Columns 1-11 (columns 1-11 on screen).

```
10 REM ★★★★★★★★★★★★★★
20 REM ★        HEX        ★
30 REM ★  BY D. R. OWEN  ★
40 REM ★      JAN '83      ★
50 REM ★★★★★★★★★★★★★★
60 CLS(0):PLAY'T255ABDGEO2AFDECGBFANDGCFEGABGFEGDFAGO3AGDL12GADB"
70 SOUND60,5
80 PRINT@13," HEX ";
90 PRINT@45, " === ";
100 PRINT@96, "DO YOU WANT INSTRUCTIONS? (Y/N)"
110 D$ = INKEY$:IF D$ = "Y" THEN 140
120 IF D$ = "N" THEN 360
130 IF D$<>"Y" OR D$<>"N" THEN 100
140 ' INSTRUCTIONS
150 SOUND140,5
160 CLS(4)
170 PRINT@35,"THE RULES OF HEX"
180 PRINT@67, "================"
190 PRINT@129,"THE OBJECT OF THE GAME IS TO BRIDGE THE PLAYING BOARD WITH YOUR
    COUNTERS, EG. BY TAKING TURNS, THE PERSON FIRST TO COMPLETE HIS BRIDGE ACROSS THE
    BOARD WINS. WHITE GOES FIRST. WATCH OUT FOR SOME SLY TACTICS!";
200 PRINT@480,"PRESS RETURN FOR MORE";
210 F$ = INKEY$:IF F$ = CHR$(13) THEN 230
220 IF F$<>CHR$(13) THEN 210
230 CLS(3)
240 PRINT@2,"HOW TO PLACE YOUR PIECES";
250 PRINT@34,"========================";
260 PRINT@97,"PIECES ARE PLACED ON THE BOARD BY INPUTTING THE CORRECT CO-ORDINATES
    OF THE DESIRED POSITION, EG: G,7 WOULD PLACE EITHER A WHITE OR AN ORANGE COUNTER
    ON POSITION G,7 AND SO ON";
270 FORI = 1TO5000:NEXTI
280 CLS(6)
290 PRINT@35,"HINT";
300 PRINT@67,"=====";
310 PRINT@129,"SINCE THE OBJECT OF THE GAME IS TO BUILD A BRIDGE ACROSS THE BOARD, AS
    WELL AS TRYING TO BUILD ONE YOURSELF, YOUR OPPONENT IS ALSO TRYING TO BUILD ONE
    SO A GOOD TACTIC WOULD BE TO BLOCK HIS/HER PATH.";
320 PRINT@480,"PRESS RETURN TO START";
330 R$ = INKEY$
340 IF R$ = CHR$(13) THEN 360
350 IF R$<>CHR$(13) THEN 320
360' START
370 PMODES,1
380 COLOR6,7
390 PCLS
400 SCREEN1,1
410 FOR N= 11 TO 1 STEP ~ 1
420 X= 116 - 10*N:Y0 = 12 + 8*N:Y1 = 172 - 8*N
430 I = 0 TO N
```

Continued on page 40

Open File

```
440 X1 = X + 20*I + 10:X2 = X1 + 10:X3 = X2 + 10
450 LINE(X1,Y0 - 8) - (X1,Y0 - 3),PSET
460 LINE(X1,Y1 +8) - (X1,Y1 +13),PSET
470 IF I = N THEN 530
480 LINE(X1,Y0 - 8) - (X2,Y0 -11),PSET
490 LINE(X2,Y0 - 11) - (X3,Y0 - 8),PSET
500 LINE(X1,Y1 + 13) - (X2,Y1 + 16),PSET
510 LINE(X2,Y1 + 16) - (X3,Y1 + 13),PSET
520 NEXTI
530 NEXTN
540 PRINT(0,0),6,6
550 FORI = 85TO90:PSET(14,1,5):NEXT:PSET(12,85,5):PSET(12,90,5):PSET(16,90,5)
560 PSET(20,78,5) : PSET(22,77,5) : PSET(24,78,5) : PSET(24,79,5) : PSET(22,80,5) : PSET(20,81,5) :
    PSET(20,82,5) : PSET(22,82,5) : PSET(24,82,5)
570 PSET(30,70,5) : PSET(32,69,5) : PSET(34,70,5) : PSET(34,71,5) : PSET(32,72,5) : PSET(34,73,5) :
    PSET(34,74,5) : PSET(32,75,5) : PSET(30,74,5) : PSET(34,69,5)
580 FORI = 61TO66 : PSET(44,I,5) : NEXT : PSET(40,61,5) : PSET(40,62,5) : PSET(40,63,5) : PSET(40,64,5) :
    PSET(42,64,5)
590 PSET(50,54,5) : PSET(52,54,5) : PSET(54,54,5) : PSET(50,55,5) : PSET(50,56,5) : PSET(52,56,5) :
    PSET(54,57,5) : PSET(54,58,5) : PSET(52,59,5) : PSET(50,58,5)
600 FOR = 46TO50 : PSET(60,I,5) : NEXT : PSET(62,45,5) : PSET(62,51,5) : PSET(64,50,5) : PSET(64,49,5) :
    PSET(63,48,5)
610 PSET(70,38,5):PSET(72,38,5):PSET(74,38,5):PSET(74,39,5):PSET(72,40,5):PSET(70,41,5):PSET(70,42,5):
    PSET(70,43,5)
620 PSET(82,30,5):PSET(80,31,5):PSET(80,32,5):PSET(84,31,5):PSET(84,32,5):PSET(80,33,5):PSET
    (80,34,5):PSET(80,35,5):PSET(84,34,5):PSET(84,35,5):PSET(82,36,5)
630 FORI = 22TO26:PSET(94,I,5):NEXT:PSET(92,21,5):PSET(90,22,5):PSET(90,23,5):PSET(92,24,5):PSET
    (92,27,5)
640 FORI = 15TO18:PSET(100,I,5):PSET(104,I,5):NEXT:PSET(102,14,5):PSET(102,19,5)
650 PSET(112,7,5):PSET(112,8,5):PSET(112,10,5):PSET(112,11,5)
660 FORI = 100TO105:PSET(10,I,5):PSET(14,I,5):NEXT:PSET(12,99,5):PSET(12,102,5)
670 FORI = 106TO111 : PSET(20,I,5) : NEXT : PSET(22,106,5) : PSET(24,107,5) : PSET(22,108,5) : PSET
    (24,109,5) : PSET(24,110,5) : PSET(22,111,5)
680 FORI = 115TO118:PSET(30,I,5):NEXT:PSET(32,114,5):PSET(32,119,5):PSET(34,115,5):PSET(34,118,5)
690 FORI = 123TO128:PSET(40,I,5):NEXT:FORI = 124TO127:PSET(44,I,5):NEXT:PSET(42,123,5):
    PSET(42,128,5)
700 FORI = 131TO136 : PSET(50,I,5) : NEXT : PSET(52,131,5) : PSET(54,131,5) : PSET(52,133,5) :
    PSET(52,136,5) : PSET(54,136,5)
710 FORI = 139TO144:PSET(60,I,5):NEXT:PSET(62,139,5):PSET(64,139,5):PSET(62,142,5)
720 FORI = 148TO151 : PSET(70,I,5) : NEXT : PSET(72,147,5) : PSET(74,148,5) : PSET(72,152,5) :
    PSET(74,151,5) : PSET(74,150,5) : PSET(74,152,5) : PSET(72,150,5)
730 FORI = 155TO160:PSET(80,I,5):PSET(84,I,5):NEXT:PSET(82,157,5)
740 FORI = 163TO168:PSET(92,I,5):NEXT
750 FORI = 171TO175:PSET(104,I,5):NEXT:PSET(102,171,5):PSET(102,176,5):PSET(100,175,5)
760 FORI = 179TO184 : PSET(110,I,5) : NEXT : PSET(114,179,5) : PSET(114,180,5) : PSET(112,181,5) :
    PSET(112,182,5) : PSET(114,183,5) : PSET(114,184,5)770 COLOR8,6
780 LINE(5, 108)—(5,186),PSET
790 LINE(5,108)—(102,186),PSET
800 LINE(5,186)—(102,186),PSET
810 PAINT(6,115),8,8
820 COLOURS,6
830 LINE(150,186)—(250,186),PSET
840 LINE(150,186)—(250,108),PSET
850 LINE(250,108)—(250,186),PSET
860 PRINT(249,185),5,5
870 LINE(5,84)—(5,5),PSET
880 LINE(5,5)—(102,5),PSET
890 LINE(102,5)—(5,84),PSET
900 PRINT(6,6),5,5
910 COLOR8,6
920 LINE(150,5)—(250,5),PSET
930 LINE(250,5)—(250,84),PSET
940 LINE(250,84)—(150,5),PSET
950 PRINT(249,6), 8,8
960 FORI = 165TO174:PSET(194,I,8):PSET(204,I,8):PSET(210,I,8):NEXT
```

```
970 FORI=0TO1
980 PSET(196,169 + I):PSET(198,169 + I,8):PSET(200,169 + I,8):PSET(202,169 +I,8):NEXT
990 FORI =  0TO1:PSET(212,165 + I,8):PSET(214,165 + I,8):PSET(216,165 + I,8):PSET(212,169 +
    I,8):PSET(214,169 + I,8):PSET(212,173 + I,8):PSET(214,173 + I):PSET(216,173 + I,8):PSET(218,173 +
    I,8):NEXT
1000 PSET(224,165,8) : PSET(224,166,8) : PSET(226,167,8) : PSET(226,168,8) : PSET(228,16,9,8) : PSET
     (228,170,8) : PSET(230,167,8) : PSET(232,166,8) : PSET(228,16,9,8) : PSET(228,170,8) :
     PSET(230,168,8) : PSET(230,167,8) : PSET(232,166,8) : PSET(232,16,5,8) : PSET(226,171,8) :
     PSET(226,172,8) : PSET(224,173,8) : PSET(224,174,8) : PSET(230,17,1.8
1010 PSET(230,172,8):PSET(232,173,8):PSET(232,174,8)
1020 GOSUB1140
1030 GOSUB1320
1040 IF PPOINT(X,Y) =7 THEN 1060
1050 SOUND10,5:GOTO1020
1060 PRINT(X,Y),5,6
1070 GOSUB1140
1080 GOSUB1320
1090 IF PPOINT(X,Y) = 7 THEN 1110
1100 SOUND150,5:GOTO1070
1110 PRINT(X,Y),8,6
1120 GOTO1020
1130 GOTO1130
1140 A$ = INKEY$:IF A$ = "" THEN 1140
1150 B$ = INKEY$:IF B$ = "" THEN 1150
1160 IF B$ = CHR$(58) THEN B = 11:GOTO1200
1170 IF B$><"0" THEN 1190
1180 B = 10:GOTO1200
1190 B = VAL(B$)
1200 IF A$ = "A" THEN A = 1
1210 IF A$ = "B" THEN A = 2
1220 IF A$ = "C" THEN A = 3
1230 IF A$  = "D" THEN A= 4
1240 IF A$ = "E" THEN A = 5
1250 IF A$ = "F" THEN A = 6
1260 IF A$ = "G" THEN A = 7
1270 IF A$ = "H" THEN A = 8
1280 IF A$ = "I" THEN A= 9
1290 IF A$ = "J" THEN A = 10
1300 IF A$ = "K" THEN A= 11
1310 RETURN
1320 X = 7 + 10★(A + B):Y = 94 + 8★(A −B)
1330 RETURN
```

Basis of a text editor

Writing a program to read, edit and record a data file is straightforward, but if the same could be done for a program file then Dragon users would be able to move lines around, merge and append programs for instance. In short a text editor.

This program shows how to read in program into an array, which can then be manipulated as required.

First you must record the program to be read in ASCII format using *Csave* " name ",A. Don't worry about the unusual noises emerging from the Dragon during this operation. Then the following program will read it into array F$, a line to each element.

To edit the program, just deal with each array element as if it were a string. To save the edited program, just write it to the cassette as if it were a data file, ie *Print* −1,F$(1),F$(2), . . .

Don't forget to make enough string space available with a *Clear*. This program was submitted by M. Twigger of Fleet, Hants.

```
10 CLEAR 10000
20 DIM F$(200)
30 OPEN "I",  -1," name "
40 I=0
50 IF EOF(-1) THEN 100
60 INPUT  -1,E$
70 F$(I)=E$
80 I=I+1
90 GOTO 50
100 CLOSE  -1
```

NEW



Cruising on Broadway

Behind the wheel of your machine. **Cruising on Broadway.** You look mean. People stop to stare. Suddenly a black-and-white pulls onto the Broadway behind—its light flashing and siren wailing. The chase is on!

Cruising on Broadway is a tyre-burning fast machine-code game for the 16 & 48K Spectrum. The further you can get the more difficult it becomes—cash prizes are being offered each month for the highest scores.

£4.95*

Blind Alley

A deadly duel, deep in space. **Blind Alley** is what passes for sport in the year 3017. Your craft must use strategy to out-wit and corral the opponent—in this crazy development of the pursuit games of the late 2600s. After each round in the enclosure game more players are sent against you to test your skill and agility. **Blind Alley** is the machine-code action of the future—now! For the 16 & 48K Spectrum.

£4.95*

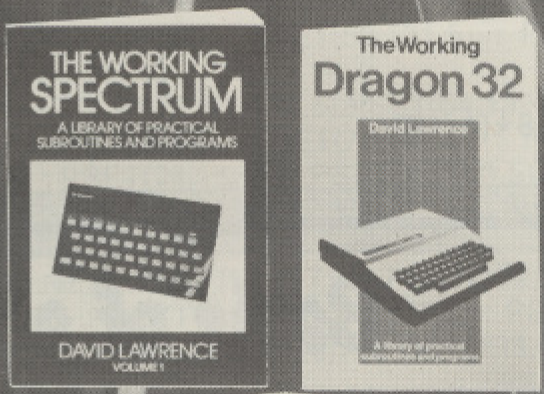
2 cassettes & 2 books from Sunshine

The Working Spectrum

(228 page book)

The first well-documented collection of serious programs for the ZX Spectrum, Programs include a Basic Renumber which can handle Gotos and Gosubs, a character dictionary, a file-handler, several utility programs and a few games. Each program is built up out of re-usable subroutines.

£5.95*



The Working Dragon 32

This is a companion volume to The Working Spectrum and includes several new features such as a Text Editor and a Music Composer and Editor. Other programs help you use the Dragon 32 as an accountant, for more advanced high resolution graphics, as a home tutor for education and for playing games. Each of the programs and subroutines is explained line by line to help develop your own programming skills.

£5.95*

***Submit *your* program to Sunshine and your name could be here next month.**

SUNSHINE

*Available through W.H. Smith's and computer dealers.

Trade/dealer enquiries welcome

Please send me

☐ **The Working Spectrum** at £5.95 each

☐ **Cruising on Broadway** at £4.95 each

☐ **The Working Dragon 32** at £5.95 each

☐ **Blind Alley** at £4.95 each

I enclose a cheque/ postal order for £ ________

payable to **Sunshine Books**
19 Whitcomb Street, London WC2 7HF.

Name ________

Address ________

Signed ________

We can normally deliver in four to five days.

Pacman

This is a Pacman-type game which has been translated from Sinclair Basic to Dragon Basic. The program takes up about 3K of Ram.

The object of the game is to eat as many of the power pills '*' and as many pills '.' as possible before the ghost eats you. You or the ghost cannot go through walls. You can chose between two speeds fast or slow. This is asked for at the start of the game. If you get 5,000 or 10,000 points you get an extra life.

This program uses sound to tell you if you have any power or if you eat a power pill or if you eat a pill.

The program also has high scores and if you have the highest score it asks for your name. Then at the end of every game and if no one beats your score your name is printed up.

The program uses a Peek to find out if you have hit anything and the same with the ghost.

The program should be bug-free and I hope that all Dragon owners have a good time. A tip for Dragon owners:

If you are saving a program and you wish to find out if it is saved correctly then if you rewind your tape and then type in Cload 'verify' then this checks if the program is saved correctly.

If you get a '?IO' error then it is not saved correctly. If it is OK then press 'reset' to get back to your program. This does not go over the program in memory — that is if you don't have a program called 'verify' on the tape. This program was submitted by B. Watson of Edinburgh.

```
10  REM  (C) 1/2/83 BRIAN WATSON
20 CLS
30  INPUT "FAST OR SLOW ";A$
40  IF A$ = "F" OR A$ = "FAST" THEN POKE 65495,0
50  IF A$ = "S" OR A$ = "SLOW" THEN POKE 65494,0
60 G$ = "G": REM  INVERSE 'G'
70 SS = 0:CO = 0
80 LL = 4
90 CLS4
100 L = L + 1
110 W$ = "."
120  RESTORE
130  FOR N = 0 TO 485 STEP 32: READ A$: PRINT @N,A$;: NEXT
140 M = 465:G = 335
170  PRINT @19,"POWER="; INT (PO);: PRINT @51,"LIVES="LL;: PRINT @51 + 32,"SCORE=";SS;
180  PRINT @M,"@";:MM = M
190  IF PO < 4 THEN G$ = "G": REM INVERSE'G'
200  IF PO <  > 0 THEN PO = PO - .5
210  IF PO > 4 THEN G$ = "G": REM NORMAL 'G'
220  IF PO <  > 0 THEN PLAY"O4L200G"
230  IF PO = 1 THEN PLAY"O1L200AO2L200AO3L200AO4L200A"
240  PRINT @G,G$;:GG=G
250  IF SS = 5000 THEN LL = LL + 1:SS = SS + 1
260  IF SS = 10000 THEN LL = LL + 1:SS = SS + 1
270 Q$ = INKEY$
280  IF Q$ = "I" THEN M = M - 32
290  IF Q$ = "M" THEN M = M + 32
300  IF Q$ = "J" THEN M = M - 1
310  IF Q$ = "K" THEN M = M + 1
320  IF  PEEK (1024 + M) = 9 THEN M = MM
330  IF  PEEK (1024 + M) = 106 THEN PLAY"O4L25A":PO = 50:SS = SS + 10
340  IF  PEEK (1024 + M) = 110 THEN PLAY"O2L250BL250DL250A":CO = CO + 1:SS = SS + 1
360  IF CO = 175 OR CO = 174 THEN CO = 0: GOTO 100
370  GOSUB 400
380  GOTO 170
390 M = MM: GOTO 280
400  REM  MOVE GHOST
410 RX =  RND (3)
420 RY =  RND (3)
430  IF RX = 1 THEN G = G + 1
440  IF RY = 1 THEN G = G - 32
450  IF RX = 3 THEN G = G - 1
460  IF RY = 3 THEN G = G + 32
470  PRINT @MM," ";
480  PRINT @GG,W$;
490 W$ = " "
500  IF G = M OR GG = M OR GG = MM OR G = MM THEN 560
520  IF  PEEK (1024 + G) = 9 THEN G = GG
530  IF  PEEK (1024 + G) = 110 THEN W$ = "."
540  IF  PEEK (1024 + G) = 106 THEN W$ = "*"
550  RETURN
560  IF PO <  > 0 THEN 690
570 PLAY"L100AL100BL50C"
580 LL = LL - 1: IF LL =  - 1 THEN 600
590  PRINT @MM," ";: GOTO 140
600  PRINT "HARD LUCK YOU'RE TOO SLOW","HA! HA! HA!"
610  FOR Y = 1 TO 10
620 PLAY"L200ABCDE-FGL200AL200BL200CO2DO2E"
630  NEXT
640 CLS
650  IF SS < HS THEN 670
660  IF SS > HS THEN HS = SS: INPUT "WHAT IS YOUR NAME ";N$
670  PRINT " THE HIGHEST SCORE WAS     "HS" BY "N$
680  FOR I = 1 TO 5500: NEXT I: GOTO 10
690  REM  EATEN GHOST
700 PO = 0
710 SS = SS +  RND (4) * 50
720 G = 335
730 PLAY"L50AL100BL150CL200D"
740  GOTO 170
750  REM  MAZE...ALL LETTER "I" A
```

Continued on page 45

Classified

COMPUTER CORRESPONDENCE COURSES

LEARN TO MASTER THE DRAGON

A complete course in BASIC programming which is designed specifically for the DRAGON micro-computer. It includes word processing, file-management, stock-control, games, and educational applications. Suitable for the complete beginner, hobbyist, teacher, and business person.

Cost of Course
Including computer: **£262.40** (incl)
Excluding computer: **£89.50** (incl)
Price per Module: **£8.95** (incl)

For full details write enclosing sae to:

COMPUTER CORRESPONDENCE COURSES
Churchway, West Ilsley
Nr Newbury
BERKS RG16 0AH
Tel (06328) 336

DRAGON 32 MERGE PROGRAMS

Programs can be fully merged on the Dragon 32. For full details and instructions send £1 plus sae to **G. Grimwood, 'Allenwoods', 105 Conway Road, Colwyn Bay, Clwyd.**

CLASSIFIED ADVERTISING RATES

Line by line: 20p per word, minimum 10 words.

Semi-display: £5 per single column centimetre, minimum length 2 cm. (Please supply artwork as PMT, or supply rough setting instructions.)

Conditions: All copy for Classified section must be pre-paid. Cheques and postal orders, made payable to *Dragon User*, should arrive at least two weeks before the publication date.

If you wish to discuss your ad, please ring Simon Langston on 01-930 3840.

KOPIKAT

Dragon 32 Owners

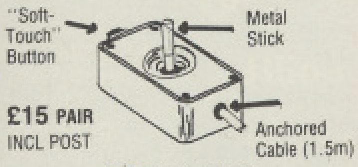
Now available: a KopiKat duplicating tape will copy most of your own m/c code tapes — even double loaders — £15.

KOPIKAT, 2 Bell Lane
Narborough, Leicestershire

PROTECT YOUR DRAGON with a fitted PVC dust cover, only £3.95 inclusive. Sherborne Designs, 'Victory House', 8a The Rank, North Bradley, Trowbridge, Wilts. Tel: 02214 4425. Access and Barclaycard. Covers for all computers, printers, monitors, etc available. Dealer enquiries welcome.

TALON

DRAGON/TANDY JOYSTICKS



£15 PAIR
INCL POST

EXTENSION LEADS
First Metre £2 and £1/m thereafter
104 Woodbridge Rd, Rushmere
St Andrew, Ipswich, Suffolk IP4 5RA

AT LAST

AN ATTRACTIVE DUST COVER, CUSTOM-MADE TO FIT YOUR DRAGON. COLOUR CO-ORDINATING TO MATCH YOUR MICRO, SCREEN PRINTED WITH "DRAGON".

(Acorn have already officially approved our BBC cover)

ONLY £3.25 INC P&P

TRADE ENQUIRIES WELCOME

COTSWOLD COVERS, PARK HILL
HOOK NORTON, OXON

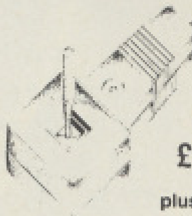
Mr Thomlinson
0608 737472

WANTED URGENTLY

Arcade type games for micros. ZX81, Spectrum, VIC 20, Dragon and Oric. We pay top royalties or buy outright. Any good games programs. National mail order and dealer sales ensure maximum reward for your work.

Reply immediately to:
NORTHWISH,
GROUND FLOOR,
RALLI BUILDING,
STANLEY STREET,
MANCHESTER 3
or phone 061-832 9143

DRAGON 32 JOYSTICKS



ONLY
£14.95
plus 75p p&p

Top quality, easy to handle joysticks, for the enthusiastic games player.

Available from

CASCOM
67 ELSTOW ROAD,
BEDFORD MK42 9NT

Here's my classified ad.

(Please write your copy in capital letters on the lines below.)

£0.20	£0.40	£0.60	£0.80
£1.00	£1.20	£1.40	£1.60
£1.80	£2.00	£2.20	£2.40
£2.60	£2.80	£3.00	£3.20
£3.40	£3.60	£3.80	£4.00
£4.20	£4.40	Please continue on a separate sheet of paper	

I make this.............................. words, at 20p per word so I owe you £..............................

Name..

Address..

....

....

Telephone..

Please cut out and send this form to: Classified Department, *Dragon User*, Hobhouse Court, 19 Whitcomb Street, London WC2

```
      RE IN INVERSE IN THE MAZE

760  DATA  "IIIIIIIIIIIIIIIIIII"
770  DATA  "I*...............*I"
780  DATA  "I..I..I..I..I..I..I"
790  DATA  "I..I..I..I..I..I..I"
800  DATA  "I..I..I..I..I..I..I"
810  DATA  "I.................I"
820  DATA  "I..IIII.III.IIII..I"
830  DATA  "I.................I"
840  DATA  "I.*I..I..I..I.*I..I"
850  DATA  "I..I..I..I..I..I..I"
860  DATA  "I.................I"
870  DATA  "I..IIII.III.IIII..I"
880  DATA  "I.................I"
890  DATA  "I.I...IIIIIII...I.I"
900  DATA  "I.................I"
910  DATA  "IIIIIIIIIIIIIIIIIII"
```

Character generator

This program allows you to design 8 × 8 characters for the high resolution screen. When the program is run a large 8 × 8 grid appears on the screen.

In the top left hand corner is a flashing cursor, this cursor can be moved around the screen by the use of the four arrow keys. If you wish to fill any of the squares press the shift key as well.

When you have finished your design move the cursor out of the grid or into a blank square then press key S. The five arrays you can use (A to E) are plotted on the right side of the grid, if the array is empty a black square is shown. Any characters used can be stored on tape and then you can go back to them later for further use.

You can use other modes and colours by altering line 9030 but I have found that mode 4 is the most clear to use. This program was submitted to us by C. Roberts who lives in Bootle, Merseyside.

Program notes

Lines

9000-9090	Sets up arrays and draws high resolution screen.
9100-9320	Controls the cursor movement and reads the keyboard.
9330-9390	Plots character at bottom of screen.
9400-9440	Selects array to be used.
9450-9510	*Get*s character from bottom of screen and stores this in the array selected then returns to high res screen.

```
9000 REM DRAGON
      CHARACTER
      GENERATOR
9010 REM CLIVE ROBERTS 1982
9020 DIM Z(10,10):DIM A(8,8):DIM B(8,8):DIM
     C(8,8):DIM D(8,8):DIM E(8,8)
9030 PMODE 4:SCREEN 1,1:COLOR 0,1:PCLS
9040 DRAW "BM0,0R11D88R11U88R11D88R11
     U88R11D88R11U88R11D88R11U88L88
     D88R88U11L88U11R88U11L88U11R88
     U11L88U11R88U11L88"
9050 PUT(150,0) - (157,7),A,PSET
9060 PUT(170,0) - (177,7),B,PSET
9070 PUT(190,0) - (197,7),C,PSET
9080 PUT(210,0) - (217,7),D,PSET
9090 PUT(230,0) - (237,7),E,PSET
9100 VE = 1:HO = 1
9110 GOTO 9160
9120 FOR Y = 1 TO 50
9130 IF PEEK(337) < 255 THEN 9220
9140 NEXT Y
9150 RETURN
9160 PUT(HO,VE) - (HO + 9,VE + 9),Z,PSET
9170 GOSUB 9120
9180 PUT(HO,VE) - (HO + 9,VE + 9),Z,PRESET
9190 GOSUB 9120
9200 GOTO 9160
9220 PUT(HO,VE) - (HO + 9,VE + 9),Z,PRESET
9230 IF PEEK(337) = 223 AND PEEK(344) = 223
     AND HO < 77 THEN HO = HO + 11
9240 IF PEEK(337) = 223 AND PEEK(343) = 223
     AND HO > 1 THEN HO = HO - 11
9250 IF PEEK(337) = 223 AND PEEK(342) = 223
     AND VE < 77 THEN VE = VE + 11
9260 IF PEEK(337) = 223 AND PEEK(341) = 223
     AND VE > 8 THEN VE = VE - 11
9270 IF PEEK(337) = 159 AND PEEK(344) = 223
     AND HO < 88 THEN PUT(HO,VE) - (HO +
     9,VE + 9),Z,PSET:HO = HO + 11
9280 IF PEEK(337) = 159 AND PEEK(343) = 223
     AND HO > 1 THEN PUT(HO,VE) - (HO + 9,VE
     + 9),Z,PSET:HO = HO - 11
9290 IF PEEK(337) = 159 AND PEEK(342) = 223
     AND VE < 88 THEN PUT(HO,VE) - (HO
     +9,VE + 9),Z,PSET:VE = VE + 11
9300 IF PEEK(337) = 159 AND PEEK(341) = 223
     AND VE > 1 THEN PUT(HO,VE) - (HO + 9,VE
     + 9),Z,PSET:VE = VE - 11
9310 IF INKEY$ = "S" THEN 9330
9320 GOTO 9160
9330 HO = -1:VE = -1
9340 FOR Y = 5 TO 82 STEP 11
9350 VE = VE + 1:HO = 0
9360 FOR X = 5 TO 82 STEP 11
9370 HO = HO + 1
9380 IF PPOINT (X,Y) = 0 THEN PSET (HO,VE +
     150)
9390 NEXT X,Y
9400 CLS:SCREEN 0,1
9410 PRINT"SELECT CHARACTER A TO E"
9420 IF INKEY$ > " " THEN 9420
9430 A$=INKEY$:IF A$<"A"OR A$>"E"
     THEN 9430
9440 Q = (ASC(A$) -64)
9450 ON Q GOSUB 9470,9480,9490,
     9500,9510
9460 GOTO 9030
9470 GET(1,150) - (8,157),A,G:RETURN
9480 GET(1,150) - (8,157),B,G:RETURN
9490 GET(1,150) - (8,157),C,G:RETURN
9500 GET(1,150) - (8,157),D,G:RETURN
9510 GET(1,150) - (8,157),E,G:RETURN
```

Drawing

The line function on the Dragon is a very useful feature; however, it is limited to drawing a line from A to B, or else use the Draw command.

Here is a program that you can feed into the Dragon, then kick it around and have hours of fun. For example clock fingers moving at any speed, printing points of a circle or oval. It can be used in all kinds of programs. This program was submitted by W. Slater of Brampton, Chesterfield, Derbyshire.

Line 20 defines pi.

The figure 30 in line 50 is half the number of lines or points to be drawn in the circle or oval.

The figure 80 in lines 60 and 70 gives the size of the circle or dimensions of an oval.

```
  10 P MODE 4,1: SCREEN1,1: PCLS.
  20 DEF FN R (X) = 4.0 * ATN (1.0)
  30 FOR T = 0 TO 10000
  40 TIMER = 0
  50 A = T/30 * FN R (A)
  60 X = 128 + 80 * COS (A)
  70 Y = 96 + 80 * SIN (A)
  80 LINE (128,96) - (X,Y),PSET
  90 IF TIMER = 48 THEN GOSUB 1000
 110 NEXT T
 999 END
1000 LINE (128,96) - (X,Y), PRESET
1100 RETURN
```

Dragonet

This program was devised when playing around with the Draw command. Careful attention must be given to the instruction in the Draw command to prevent Paint spill. It is left up to you to discover exactly what it does. This program was submitted by J. Trill who comes from Rochester, Kent.

Program notes

10-120 Draw the figure.
130-180 Movement.

```
5    REM DRAGONET BY J. TRILL
1Ø   PMODE 3,1 : SCREEN 1,Ø : PCLS3
2Ø   DRAW "BM13Ø,178 ; S16C4D2L11U2E1U3E1U3H1
     U1H1L1H1L1H1L1U1L1U4H1U1H2U1H1U2H1U2
     E2U1E6R1E1R1ØF1R1E1R5D1R2F1R4G1D2
     F5D3G1D3G2D1L1G4L2G4D1G1D1G1D2G1D5"
3Ø   DRAW "BM-15,-3Ø ; E3R3F2D1G2L1G1L1H1L1H1U1E1"
4Ø   DRAW "BM+18,+1 ; U2E2R2F1R1D1F1G2L3H3"
5Ø   DRAW "BM-1,+1Ø ; U1H2G1H1L1G1L1G1L1G1R1F1
     R1E1R2F1E1R2"
6Ø   DRAW "BM+1,-1Ø ; H1G1D1F1E1U1 ; BM-4,Ø; H1G1
     D1F1E1U1"
65   PAINT (12Ø,16Ø),4,4
7Ø   DRAW "BM-7,-2; G2D2F1R1"
8Ø   DRAW "BM+18,-1; L1H2U2E2"
9Ø   PAINT (8Ø,65), 2,4
1ØØ  PAINT (15Ø,55),2,4
11Ø  DRAW "BM 6Ø,18Ø; C4E5F1G5L2E3"
12Ø  PAINT (62,179),4,4
13Ø  CIRCLE (83,16Ø),5,2
14Ø  PAINT (83,16Ø),2,2
15Ø  FOR I = 1 TO 5ØØ : NEXT I
155  CIRCLE (83,16Ø),5,4
16Ø  PAINT (83,16Ø),4,4
17Ø  FOR I = 1 TO 5ØØ : NEXT I
18Ø  GO TO 13Ø
```

Noughts and crosses

This program prints the noughts-and-crosses grid on the screen, and then invites you to play the game; first by typing x or o depending on which you are, and then by typing a number between 1 and 9: 3 is top right, 9 is bottom right, etc. This program is by C. Davies of Eastleigh.

Program notes

Lines	
2-10	Print headings and clear screen.
20-60	Set hi-res graphics and draws grid.
70-100	Established whether it is "x" or "o".
500-600	Established position of "x's" on grid.
700-800	Established position of "o's" on grid.
2000-2430	Draw the "x's" on the screen.
3000-3420	Draw the "o's" on the screen.

```
1 REM "noughts and crosses"
2 REM "CARL DAVIES, 1982
3 PRINT @ 128 + 11 ,"NOUGHTS'
4 PRINT @ 192 + 13 ,"AND"
6 PRINT @ 256 + 11 ,"CROSSES"
8 FOR Z = 1 TO 1300:NEXT Z
10 CLS
20 PMODE 3,1:SCREEN 1,0:PCLS
30 LINE (110,20) - (110,140),PSET
40 LINE (150,20) - (150,140),PSET
50 LINE (70,60) - (190,60),PSET
60 LINE (70,100) - (190,100),PSET
70 A$ = INKEY$
80 IF A$ = "x" THEN GOSUB 500;
90 IF A$ = "o" THEN GOSUB 700;
100 GOTO 70
500 B$ = INKEY$
510 IF B$ = "1" THEN GOSUB 2000;
520 IF B$ = "2" THEN GOSUB 2050;
530 IF B$ = "3" THEN GOSUB 2100;
540 IF B$ = "4" THEN GOSUB 2150;
550 IF B$ = "5" THEN GOSUB 2200;
560 IF B$ = "6" THEN GOSUB 2250;
570 IF B$ = "7" THEN GOSUB 2300;
580 IF B$ = "8" THEN GOSUB 2350;
590 IF B$ = "9" THEN GOSUB 2400;
600 GOTO 500
700 C$ = INKEY$
710 IF C$ = "1" THEN GOSUB 3000;
720 IF C$ = "2" THEN GOSUB 3050;
730 IF C$ = "3" THEN GOSUB 3100;
740 IF B$ = "4" THEN GOSUB 3150;
750 IF B$ = "5" THEN GOSUB 3200;
760 IF B$ = "6" THEN GOSUB 3250;
770 IF B$ = "7" THEN GOSUB 3300;
780 IF B$ = "8" THEN GOSUB 3350;
790 IF B$ = "9" THEN GOSUB 3400;
800 GOTO 700
2000 LINE (80,30) - (100,50),PSET
2010 LINE (100,30) - (80,50),PSET
2020 GOTO 70
2050 LINE (120,30) - (140,50),PSET
2060 LINE (140,30) - (120,50),PSET
2070 GOTO 70
2100 LINE (160,30) - (180,50),PSET
2110 LINE (180,30) - (160,50),PSET
2120 GOTO 70
2150 LINE (80,70) - (100,90),PSET
2160 LINE (100,70) - (80,90),PSET
2170 GOTO 70
2200 LINE (120,70) - (140,90),PSET
2210 LINE (140,70) - (120,90),PSET
2220 GOTO 70
2250 LINE (160,70) - (180,90),PSET
2260 LINE (180,70) - (160,90),PSET
2270 GOTO 70
2300 LINE (80,110) - (100,130),PSET
2310 LINE (100,110) - (80,130),PSET
2320 GOTO 70
2350 LINE (120,110) - (140,130),PSET
2360 LINE (140,110) - (120,130),PSET
2370 GOTO 70
2400 LINE (160,110) - (180,130),PSET
2410 LINE (180,110) - (160,130),PSET
2420 GOTO 70
3000 LINE (80,30) - (100,50),PSET,B
3010 GOTO 70
3050 LINE (120,30) - (140,50),PSET,B
3060 GOTO 70
3100 LINE (160,130) - (180,50),PSET,B
3110 GOTO 70
3150 LINE (80,70) - (100,90),PSET,B
3160 GOTO 70
3200 LINE (120,70) - (140,90),PSET,B
3210 GOTO 70
3250 LINE (160,70) - (180,90),PSET,B
3260 GOTO 70
3300 LINE (80,110) - (100,130),PSET,B
3310 GOTO 70
3350 LINE (120,110) - (140,130),PSET,B
3360 GOTO 70
3400 LINE (160,110) - (180,130),
     PSET,B
3410 GOTO 70
```

If you've got a technical question or problem write to Cathy Hyde, Dragon Answers, *Dragon User*, Hobhouse Court, 19 Whitcomb Street, London WC2 7HF

Dragon Answers



No joy on joysticks

I HAVE a Dragon 32 and on page 88, 89 and 90 of the manual there is a game using joysticks. It is supposed to be a battle between two ships in space. It uses the joystick to move your craft and the fire button to operate your weapon to hit the other ship. But when I press the fire button on the left joystick the right joystick's ship fires at me and vice-versa. I think that the trouble is in lines 140, 150 and 160. If it is possible can you find out if there is an error anywhere.

John Winter (Age 12), Aylesbury, Bucks.

IT APPEARS THAT you have an earlier edition of the manual with a misprint on the joystick commands. The joysticks are actually reversed from those given, ie JOYSTK (0) and (1) refer to right joystick and JOYSTK (2) and (3) to the left joystick. The simplest method of altering the program on page 88 is to reverse the scores and fire buttons. Amend lines 120, 150 and 160 as below:

```
120 CLS0: FOR Y=0 TO 1:
    PRINT @0,Z(1)::PRINT @
    26,Z (0).
150 IF P = 125 OR P = 253
    THEN F = 1:T=0:GOSUB
    200.
160 IF P = 126 OR P = 254
    THEN F = 0:T = 1:GOSUB
    200.
```

SAM explained

I WONDER if you could answer the following queries I have on computers and programming.

1) What is SAM?
2) Is there any way which I could write a program that can totally ignore commands like list and break?
3) Is it possible to damage the computer by 'POKING' a wrong number?

P Ranson, Great Moor, Stockport.

THE SAM IS a Synchronous Address Multiplexer which controls the way in which the memory works. It is programmed to work in conjunction with the Video Display Generator to control PMODES and also works in conjunction with the processor to update the RAM.

There is no simple way to write a program which ignores commands such as LIST and BREAK. You would need a good knowledge of machine code and the Basic Interpreter in order to do this.

Temporary crashing can be brought about by POKING to the wrong number, especially in the area of the SAM chip but no permanent damage should be done.

Call for software

MY SON has had a Dragon 32 for several months now but is still very disappointed at the quality of software printed in computer magazines. A friend managed to get hold of a magazine called the *Rainbow*. I would like to know if you could supply me with the name and address of the distributors in this country.

M A Connolly, Bewdley, Worcs.

THE MAGAZINE *RAINBOW* is available in this country, from Elkan Electronics, 28 Bury New Road, Prestwich, Manchester. Tel: 061-773 3965.

Programs disappear

COULD YOU please explain to me why errors start to appear, and whole lines — even programs — start to disappear on my Dragon, after a while in the higher resolutions, especially re-running a particular program? I have noticed that many of such disappearing programs do not contain PCLEAR or CLEAR statements, but I'm not sure whether this is important. Is my Dragon faulty or am I one of many with the same problem?

Dennis Lawrence, South Woodford, London.

WE ARE AT PRESENT looking into the type of problem you have encountered. Transients in the domestic power supply may cause loss of programs and corruption, as described. If this is the case then pressing RESET or switching the machine off then on again usually clears the problem. However, it may be worth while returning your machine to the original dealer for testing as you may have a fault on the Dragon itself.

Skipping data

MY DRAGON 32 will not accept a very long, supposedly Microsoft compatible program, Sea Battle, taken from Creative Computing's More Basic Computer Games. Data statements appear in a number of places and these are implemented by the use of: "RESTORE" followed by the line number of the data statement as required. The Dragon gives me a syntax error at each RESTORE.

Preliminary instructions acknowledge possible difficulty and advise: "Improvise by Using a Restore and FOR. . .NEXT with Read statements to skip over the data that should be ignored". I do not understand this advice — can you assist me please?

W A Coysh, Southwick, Brighton.

RESTORE IS USED to get to the beginning of the data and, once used, the only way to get to any particular bit of data is by reading through all data before it. For example, to get to the fifth bit of data use RESTORE followed by: FOR I = 1 TO 5 READ A$ NEXT. This will read the first five values into A$ successively, and leave the fifth one there.

Crashing at speed

ON MORE than one occasion your magazine has printed information that the speed of the Dragon 32 can be increased by POKE 65495,0. On my machine this always causes an immediate program crash; it is necessary to switch off to regain control. What am I doing wrong?

B Hayton, Willaston, South Wirral.

THE DRAGON 32 HAS been designed to work at a certain speed and the POKE mentioned causes it to run at double speed. Therefore it is working out of the original design specifications and may crash. We do not recommend the use of this POKE for the above reasons.

Solution is a loop

I HAVE got a Dragon 32 and for the past few weeks have been trying to write a certain program. It involves the user putting in a number, say five. The computer then adds five + four + three + two + one. If the number was 10, the computer would add 10 + 9 + 8 etc. I haven't had any success in writing the program and hope you can help me out.

Chris Denne, Morden, Surrey.

THE TYPE OF PROGRAM you require is made relatively simple using the FOR NEXT loop:

```
10 ANSWER = 0 : N = 0 : A =
   0.
20 CLS.
30 INPUT "NUMBER";N.
40 IF N>0 THEN 50 ELSE 10.
50 FOR A = 1 TO N.
60 ANSWER = ANSWER + A.
70 NEXT A.
80 PRINT ANSWER.
90 IF INKEY$ = "" THEN GO
   ELSE 10.
```

Competition Corner

Send your answers to Competition Corner, *Dragon User*, Hobhouse Court, London WC2 7HF

Win a printer

ONE OF THE advantages of a computer is its unflagging ability to carry out its tasks. Even when it is running a complex program, say, high level chess, the computer is always ready and willing for another game. Similarly, its capability of repeatedly carrying out simple claculations hundreds or thousands of times, is far superior to when the task is performed by a human operator, where fatigue and distraction could lead to errors. No matter how late the hour, the computer is, or should be, as bright as a button — or at least as bright as the VDU!

As an example of using the computer to run a program involving hundreds of individual calculations, let's consider some mathematical series. A series is a set of numbers, each of which bears a relationship to the preceding terms. Take this simple progression:

1/1 − 1/2 + 1/3 − 1/4 + 1/5 − 1/6 + ...

This can be easily translated into a simple program, and if the program is run and the cumulative total displayed as each successive term is added, it can be seen to be converging towards a specific value. As the series progresses, the total stabilises towards 0.693147 . . ., the natural logarithm of 2. This is one of the ways in which this value can be evaluated empirically.

Other mathematical constants can be similarly found. The series 1 + 1/1! + 1/2! + 1/3! + 1/4! + 1/5! + . . . converges on 2.718281828 . . ., the mathematical constant *e*. The exclamation marks are factorial signs. For example, 4!, or factorial 4, equals 4 * 3 * 2 * 1.

That enigmatic constant, pi, despite its irrational value, can be evaluated by means of a series. Here are just two series which produce pi.

SERIES (1):

2 * (2/1 * 2/3 * 4/3 * 4/5 * 6/5 * 6/7 * 8/7 * 8/9 . . .)

Note that the numerator and denominator increase by two, but on alternate steps.

SERIES (2):

4 * (1/1 − 1/3 + 1/5 − 1/7 + 1/9 − . . .)

It seems incredible that so simple a series as this can produce so complex a value as pi.

The series that we have considered so far are *convergent* — that is, they converge to a definite value. Another type of series is called divergent — and these increase without limit. For instance the series

1 + 2 + 3 + 4 + 5 + 6 + 7 + . . .

will do this, increasing to infinity — or, in the case of a computer, until the mathematical capacity of the machine is exceeded. Obviously, if each successive term gets larger then the series is divergent. Does this mean that if successive terms get smaller, the series will converge to a finite value? Not necessarily — consider the following:

1 + 1/2 + 1/3 + 1/4 + 1/5 + 1/6 + 1/7 + 1/8 + . . .

Here, although each term gets smaller the series does, in fact, diverge. This can be seen readily if we break the series up into sections. The two terms 1/3 + 1/4 sum to more than 1/4 + 1/4. Therefore, these two terms are greater than 1/2.

Similarly the next four terms, 1/5 + 1/6 + 1/7 + 1/8 are greater than 1/8 + 1/8 + 1/8 + 1/8. This group is therefore also greater than 1/2, as is the next group of eight terms, the next group of sixteen terms, and so on. We can now see that our final total must be greater than:

1 + 1/2 + 1/2 + 1/2 + 1/2 + 1/2 + . . .

Thus, it will diverge to infinity, although it will do so very slowly.

The rate of divergence of the series 1 + 1/2 + 1/3 + 1/4 + 1/5 + 1/6 + 1/7 + 1/8 + . . . forms the basis of this month's competition.

It can be seen that the total sum exceeds 2 when the fourth term is added (1 + 1/2 + 1/3 + 1/4). In order to exceed 3 the series has to be extended to include the eleventh term.

How many terms need to be added for the total to exceed each integer up to 10?

Prizes

THE PRIZE FOR this month's competition is donated by the software house Microdeal. The package offered is: one Epson MX80F/T III dot matrix printer for your Dragon, a word processing package and the full range of software from Microdeal.

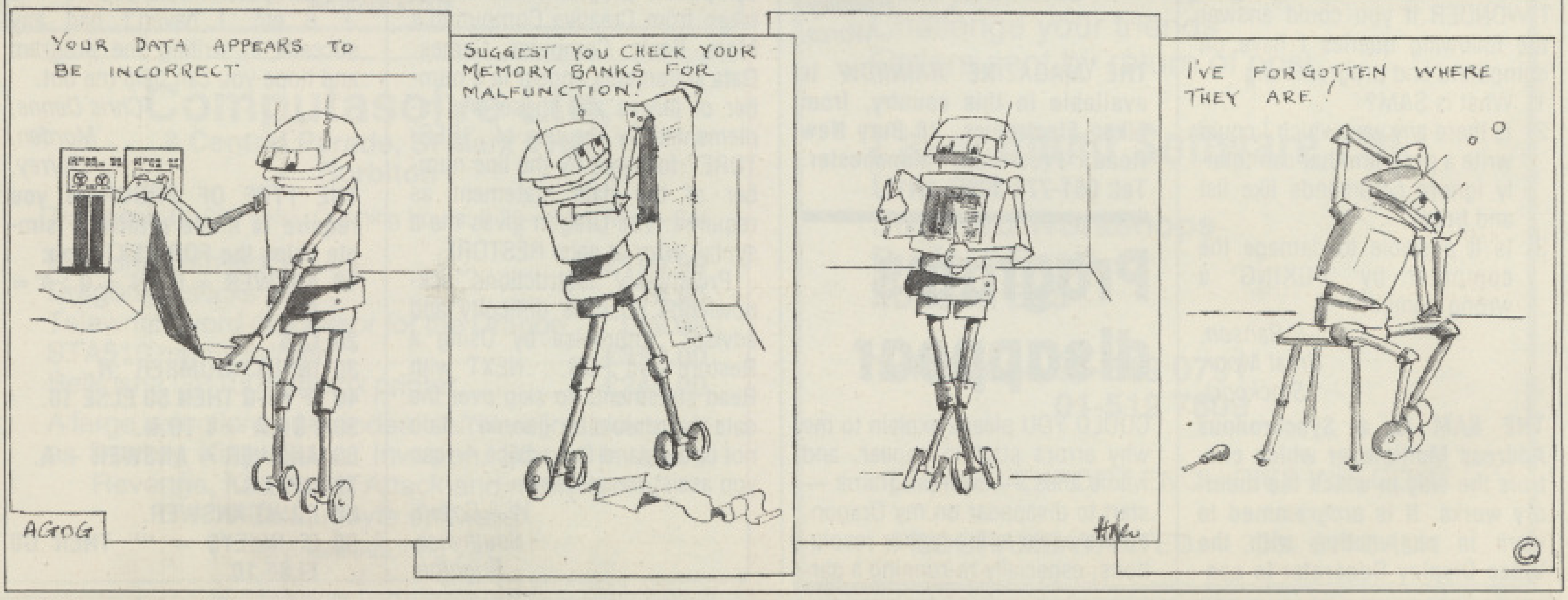
Rules

TO WIN THE software and printer, all you have to do is send in the most elegant solution to the puzzle. You must show how the puzzle can be solved with the use of a Basic program, developed on your Dragon 32 computer.

As a tiebreaker, complete the sentence below in 10 words or less:

I will use my Dragon 32 as a word processor because............................

Your entry must arrive at *Dragon User* by the last working day in May 1983. The name of the winner, and the solution to the puzzle submitted by the winner, will be printed in the July issue of *Dragon User*. You may only enter the competition once. Entries will not be acknowledged and we cannot enter into correspondence on the result.






DRAGON OWNERS
BARRELS OF FUN IN THE NEW
ANOTHER GREAT GAME FROM MICRODEAL – GUARANTEED TO KEEP YOU PLAYING FOR HOURS TRYING TO SAVE THE BEAUTY FROM THE BEAST. ROLLING BARRELS! KILLER FLAMES! LADDERS AND RAMPS TO NEGOTIATE IN THE BEST GAME EVER FOR THE DRAGON 32 HOME COMPUTER.
DONKEY KING
FROM TOM MIX
NEW
DONKEY KING
MICRODEAL
NEW
Katerpillar Attack
Planet Invasion
SPACE WAR
Defense
Scarfman
DRAGON INVADERS
ALCATRAZ II
Flipper
ESCAPE
SPACE MONOPOLY
MANSION ADVENTURE I
Jerusalem ADVENTURE 2
Williamsburg ADVENTURE 3
ULTIMATE adventure 4
All Cassettes £8 each (including VAT & Postage)
AVAILABLE FROM DRAGON 32 DEALERS NATIONWIDE OR ORDER DIRECT FROM:
MICRODEAL
41 TRURO ROAD, ST. AUSTELL, CORNWALL PL25 5JE TEL 0726-67676
Send 1st Class Stamp for FREE Software Lists for Dragon and Tandy range of Computers.
AVAILABLE FROM SELECTED Boots & John Menzies STORES
WE WELCOME ACCESS
AMERICAN EXPRESS